Université de Montréal

# Novel Multicast Protocols in Ad-Hoc Networks

Par :

Astier Eric Jean Léopold

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de

**Maîtrise ès sciences (M.Sc.)**

en Informatique

Décembre, 2008

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé

Novel Multicast Protocols in Ad-Hoc Networks

présenté par :

Astier Eric

a été évalué par un jury composé des personnes suivantes :

Dr. Esma Aimeur, President

Dr. Abdelhakim Hafid, Thesis Director

Dr. Houari Sahraoui, Jury Member

# Résumé

Un réseau ad-hoc mobile (MANET) est un réseau de nœuds mobiles qui s'organise en topologie arbitraire grâce aux liaisons sans fils entre les nœuds. Un protocole multicast permet de livrer l'information à un groupe de nœud simultanément. Il utilise les liens du réseau une seule fois afin d'atteindre plusieurs destinations. Les nœuds dans un MANET ont des quantités d'énergie limitées. Cette énergie est consommée soit en transmettant des paquets, soit en les recevant. Un protocole multicast, s'il se veut être extensible, doit équilibrer les paquets de données et de surcharge opérationnelle des paquets de contrôle (overhead). Cet équilibre doit aussi prendre en compte la consommation d'énergie et la topologie dynamique causée par les déplacements des nœuds. Rassembler les nœuds en groupes réduit le nombre de nœuds qui sont impliqués dans le routage et permet de réduire la surcharge opérationnelle.

Différentes techniques de regroupement (clustering) et de protocoles multicast ont été proposés. Les techniques de regroupement ont différents degrés de succès selon les scenarios. Certaines marchent mieux dans des situations à faible mobilité, d'autres dans des réseaux à large population et haute densité. Cependant, elles peuvent causer la mort prématurée de certains nœuds, des partitions dans le réseau et des interruptions de communication dans des situations à faible densité. Les charges de trafic ne sont pas distribuées équitablement dans un manet, des goulots d'étranglements d'énergie peuvent survenir, plus spécifiquement dans les scénarios de faible densité (la forme du réseau et la distribution des nœuds sont plus arbitraire).

Les protocoles multicast dans la littérature ont différentes méthodes pour exécuter les fonctions multicast. Certains de ces protocoles provoquent la saturation du réseau par les paquets de contrôle à cause des coûts élevés de réparation des liens. D'autres protocoles ne s'ajustent pas en fonction de la mobilité des nœuds et inondent inutilement le réseau de paquets.

D'abord, nous avons conçu un algorithme qui tient compte de la mobilité et de l'énergie résiduelle des nœuds voisins quand on sélectionne les nœuds critiques (p.ex.

chef de groupes). Puis, nous avons créé un protocole multicast qui tient compte de la mobilité et de la topologie du réseau pour réduire la latence et diminuer la surcharge opérationnelle des paquets de données (data packet overhead).

**Mots-Clefs:** MANET, Nœuds Mobiles, AODV, MAODV, Multicast, Réseau Ad-hoc, NS-2, Techniques de Regroupement Passif (Clustering), RSIDS, MOBIC, ILBH, GRIDS, Prise en Compte de l'Énergie, Mobilité, PUMA, CPUMA, MODA, ROMANT, Centralisation du Noyau.

# Abstract

A mobile ad hoc network (MANET) is a network of mobile nodes that organizes itself into an arbitrary topology by the wireless links between nodes without the presence of a wired support infrastructure. A multicast protocol allows for the delivery of information to a group of nodes simultaneously. It uses links of the network only once to reach multiple destinations. Nodes in a MANET have limited amounts of energy which is consumed by transmitting and receiving packets. A scalable multicast protocol must balance data packet delivery and overhead. This balance must contend with the power consumption and dynamic topology of moving nodes. Clustering reduces the number of nodes involved in routing and is used to reduce overhead.

Several clustering techniques and multicast protocols have been proposed. Clustering techniques have varying degrees of success in different scenarios. Some only work in low mobility scenarios, since they do not take node mobility into account when clustering. Some techniques work well in large-population-high-density networks but may cause nodes to fail, network partition and communication interruption in low density situations. Traffic loads in MANETs are evenly distributed, energy bottlenecks can happen, especially in low density scenarios (more arbitrary network shape and node distribution). Multicast protocols in the literature have very different methods for performing multicast functions. Some of the protocols cause networks to be overwhelmed by control packets due to the high costs of their link repair operations. Some protocols fail to adjust for node mobility and have unnecessary packet broadcasts.

First we designed a clustering algorithm that considers both mobility and remaining energy of neighboring nodes when selecting critical nodes (e.g. cluster heads). Second, we developed a multicast protocol that takes into account mobility and network topology to reduce latency and lower data packet overhead.

**Keywords**: MANET, Mobile Nodes, AODV, MAODV, Multicast, Ad Hoc Networks, NS-2, Passive Clustering, RSIDS, MOBIC, ILBH, GRIDS, Energy Aware, Mobility, PUMA, CPUMA, MODA, ROMANT, Centering, Core

# Table of Contents

Résumé ..... iii
Abstract ..... v
Table of Contents ..... vii
Table List ..... x
Figure List ..... xi
Abbreviation List ..... xii
Acknowledgements ..... xiv
Chapter 1: Introduction ..... 1
1.1 Mobile Ad Hoc Networks ..... 1
1.2 Multicast Protocols ..... 3
1.3 Clustering ..... 4
1.4 Motivation ..... 4
1.5 Problem Statement ..... 5
1.6 Contributions ..... 6
1.7 Thesis Organization ..... 7
Chapter 2: State of the Art ..... 8
2.1. Clustering: Definitions ..... 8
Benefits of Clustering ..... 10
Costs of Clustering ..... 10
2.2. Clustering: Schemes ..... 11
Mobility Based Metric for Clustering (MOBIC) ..... 11
Passive Clustering (PC) ..... 12
Geographically Repulsive Insomnious Distributed Sensors (GRIDS) ..... 12
Interval-Based Load Balancing Heuristics (ILBH) ..... 13
2.3. Multicast Routing: Definitions ..... 14
2.4. Multicast Routing: Protocols ..... 15
Multicast Ad Hoc On-Demand Distance Vector (MAODV) ..... 15
Robust Multicasting in ad hoc networks using trees (ROMANT) ..... 17

Protocol for Unified Multicast Announcements (PUMA) ........................................ 17
Multicasting on Directional Antennas (MODA) ........................................ 18
2.5. Conclusion ........................................ 19
Chapter 3: Energy and Mobility Aware Clustering Technique for Multicast Routing Protocols in Wireless Ad Hoc Networks ........................................ 21
Background ........................................ 21
Abstract ........................................ 22
I. Introduction ........................................ 23
II. Related Work ........................................ 25
III. RSIDS Protocol Description ........................................ 27
IV. Simulation Model and Methodology ........................................ 33
V. Conclusion ........................................ 41
References ........................................ 42
Chapter 4: Centered and Robust Multicast Routing in Mobile Ad Hoc Networks ........................................ 44
Background ........................................ 44
Abstract ........................................ 45
I. Introduction ........................................ 45
II. Related Work ........................................ 47
III. CPUMA ........................................ 49
A. Overview ........................................ 49
B. The Multicast Announcement ........................................ 49
C. Connectivity and Source Lists ........................................ 51
D. Core Election and Centering ........................................ 53
E. Mesh Establishment and Maintenance ........................................ 54
F. Forwarding Multicast Data Packets ........................................ 55
IV. Simulation Model and Methodology ........................................ 57
A. Metrics ........................................ 60
B. Scenarios ........................................ 60
C. Results ........................................ 62

V. Conclusion ........ 66
References ........ 68
Chapter 5: Conclusion and Future Work ........ 69
5.1. Future Work ........ 71
Bibliography ........ 72

# Table List

**Chapter 2**

Table 1 – Clustering Algorithms .......................................................... 19

Table 2 – Multicast Protocols ............................................................ 20

**Chapter 3**

Table 1 - Pseudo Code ...................................................................... 29

Table 2 - Node Deaths and Network Death ............................................ 40

**Chapter 4**

Table 1 - CPUMA Multicast Announcement Format .................................. 50

Table 2 – CPUMA Header ................................................................... 51

Table 3 - Connectivity List at Mesh Member 6 ......................................... 53

Table 4 - Source List at Mesh Member 6 ................................................ 53

Table 5 - CPUMA Pseudo-code ............................................................ 56

Table 6 - PUMA vs. CPUMA statistics .................................................. 59

Table 7 - Simulation Parameters ......................................................... 61

# Figure List

## Chapter 2

Figure 1 - Unicast and Multicast..... 3
Figure 2 - MAODV Route Requests and Route Replies ..... 16

## Chapter 3

Figure 1 - GRIDS Clustering..... 32
Figure 2 - RSIDS Clustering..... 33
Figure 3 - Total Reception Ratio (5 to 20, 6m)..... 35
Figure 4 - Total Reception Ratio (10 to 20, 10m)..... 36
Figure 5 - Periodic Packet Reception Ratio (5 to 20, 6m)..... 37
Figure 6 - Periodic Packet Reception Ratio (10 to 20, 10m)..... 38
Figure 7- Remaining Energy (5 to 20, 6m)..... 39
Figure 8 - Remaining Energy (10 to 20, 10m)..... 39
Figure 9 - Overhead/Data Ratio (5 to 20, 6m)..... 41

## Chapter 4

Figure 1 - Mesh broadcasting Multicast Announcements..... 52
Figure 2 - PUMA showing Data Packet Overhead..... 58
Figure 3 - CPUMA after Centering the Core..... 59
Figure 4 - Packet Delivery Ratio..... 62
Figure 5 - Control Overhead..... 63
Figure 6 - Data Packet Overhead..... 64
Figure 7 – Latency..... 65
Figure 8 - Traffic (Kbytes)..... 66

# Abbreviation List

| Acronym | Description |
| --- | --- |
| AODV | Ad-Hoc On Demand Distance Vector |
| CH | Cluster Head Node |
| CHR | Cluster Head Ready Node |
| CPUMA | Centered Protocol for Unified Multicasting through Announcements |
| GPS | Global Positioning System |
| GRIDS | Geographically Repulsive Insomnious Distributed Sensors |
| GW | Gateway Node |
| ILBH | Interval-Based Load Balancing Heuristics |
| MA | Multicast Announcement |
| MACT | Multicast Route Activation Packet |
| MANET | Mobile Ad-hoc Network |
| MAODV | Multicast Ad-Hoc On Demand Distance Vector |
| MOBIC | Mobility Based Metric for Clustering |
| MODA | Multicasting on Directional Antennas |
| NS-2 | Network Simulator 2 |
| ON | Ordinary Node |
| PC | Passive Clustering |
| PDR | Packet Delivery Ratio |
| PUMA | Protocol for Unified Multicasting through Announcements |
| ROMANT | Robust Multicasting in Ad hoc Networks using Trees |
| RREQ | Route Request |
| RREP | Route Reply |
| RSIDS | Restful Stability based Insomnious Distributed Sensors |
| TTL | Time To Live |

*To my parents.*

*To my grand-mother.*

*To my sister.*

*To my brother.*

# Acknowledgements

First of all, I would like to express my gratitude to my research director Dr. Abdelhakim Hafid who took time out of his busy schedule and personal time to offer advice, read and comment my articles. I thank him for being patient and understanding.

Thanks to my family, especially my mother Marie-Thérèse Astier and father Roland Astier for their love, generosity and trust in my abilities. Additionally, I would like to thank my sister Hélène Astier and my brother Guillaume Astier for all their encouragements.

Additionally, I thank Sidney Santos Doria, an alumnus from Federal University of Campina Grande for providing the initial NS-2 code for PUMA.

Finally, I give my gratitude to everyone that supported me especially my friends and the University of Montreal administrative staff.

# Chapter 1: Introduction

## 1.1 Mobile Ad Hoc Networks

Technological advanced in wireless data communication devices has led to their widespread use. Every year prices decline and these devices improve. Newer wireless devices are capable of data rates greater than some wired infrastructure and have a range of approximately 70 meters. Turn on the wireless card of any computer near an apartment building and you are bound to find multiple wireless networks. Most of these networks are configured so wireless devices connect to a wireless router which enables communication between nodes. If a device is not within range of the router it will not be part of the network. There is an alternative way to create a wireless network, the ad-hoc network. In an ad-hoc network, wireless devices connect and communicate with each other without the use of a base station or any kind of pre-existing wireless infrastructure.

Mobile ad-hoc networks (MANETs) are infrastructure-less, dynamically reconfigurable wireless networks. In order for a MANET to operate, hosts have to be willing to serve double duty as routers. Nodes participating in a MANET have to forward, packets from the source to the destination. In essence the nodes become the infrastructure. MANETs have the advantages of: 1) being created as needed, 2) fault tolerance, since the loss of any one node may not impact the network if another route can be found and 3) unconstrained connectivity since it is not limited to the range of the wireless router.

Typical applications in a MANET require the nodes to work as a group. Much interest is focused on applications sharing a close degree of collaboration between nodes, such as disaster recovery, crowd control, search and rescue, coordinated task scheduling or several battlefield communication schemes where no infrastructure-based topology is available. The main benefit of a MANET is its ease of deployment and support for communication between mobile users from anywhere. Besides data and information sharing in difficult terrains, an increasing demand for commercial 'community-centric' applications like multiplayer gaming through handheld portable devices is emerging. Other applications, such as sharing information in classrooms, conference rooms, metro-areas and much more are in demand as well.

Since nodes forward packets for each other without a wired or wireless infrastructure, a routing protocol is needed to make routing decisions. Deciding the correct route for a packet to take is made more difficult because nodes are free to move around. The route just taken may no longer exist if a node moves out of range of the others along that route. Any node along the route may also simply run out of energy and stop transmitting, or be so overloaded that it no longer has the ability to forward any more packets.

Since some of the applications, MANETs are asked to handle, require nodes to work in a group, nodes must be able to route data to a group of other nodes. A node may send the same data to the group of nodes one at a time, but it would be a terribly inefficient way to do it. This routing scheme is called unicast and basically means a transmission to a single destination. A node required to transmit data to a group of 5 other nodes would have to send 5 separate transmissions. These would be forwarded up to five times by nodes along the routes to the five destinations. A more efficient way to communicate with a group is to use a multicast routing scheme. In a multicast transmission a node forwards data to a group of nodes simultaneously going over links in the network just once. The data packet transmitted is forwarded only once by any node along the routes to the five destinations. The path splits only when it is necessary in order to reach all group members. Figure 1 below illustrates the two routing schemes. Efficiency is important in MANETs because mobile nodes are normally un-tethered to a power source. In scenarios such as search and rescue missions in a disaster area or combat duty, nodes must make do with the battery energy available to the device.

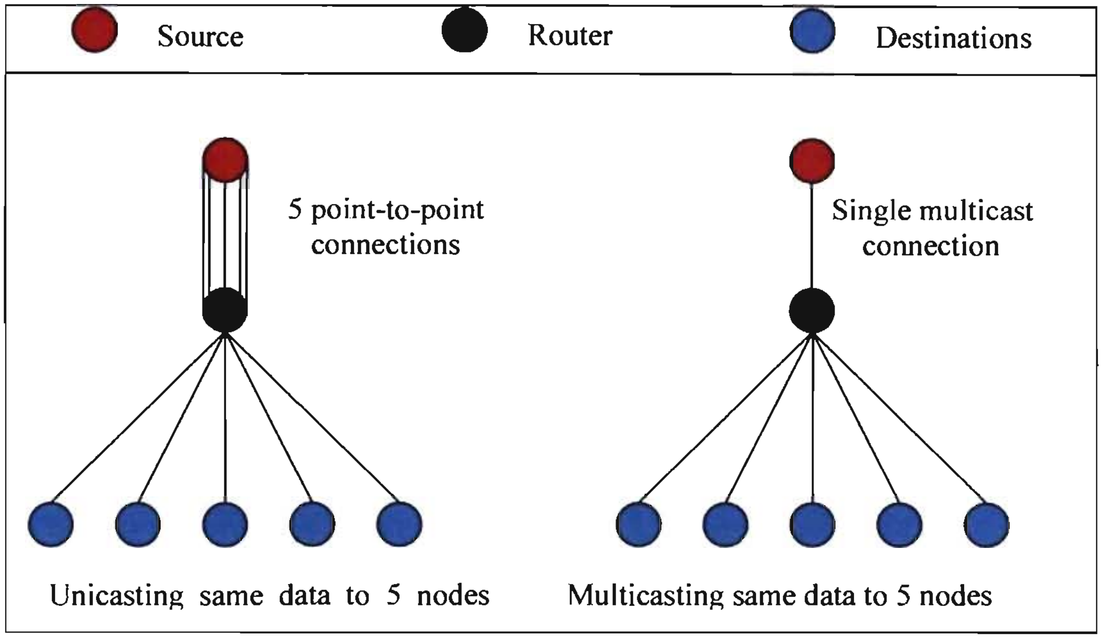


Figure 2 - Unicast and Multicast

## 1.2 Multicast Protocols

A multicast protocol is a set of rules that govern the routing decisions of multicast transmissions. As a general rule, one of the major goals in designing multicast protocols for wireless networks is to reduce unnecessary packet transmissions to nodes outside the multicast group. Many methods have been produced for multicast transmission in MANETs. The four general categories of multicast protocols are Meshed-based, Tree-based, Hybrid and Stateless.

Mesh-based protocols transmit data packets over more than one link, trading redundancy in data transmission for high robustness. Pure flooding is an extreme form of a mesh-based protocol where packets are forwarded by all nodes. Tree-based protocols create multi hop routes that trade robustness for efficiency and reduce data packet overhead. This robustness/efficiency tradeoff is a key issue in mobile ad hoc network multicasting. Hybrid protocols combine meshes and trees in an effort to enhance the positives of each method. Stateless protocols are meant for small multicast groups only and are not suited for large MANETs. In fact, most of these protocols have difficulties scaling when faced with the control and data packet overhead of hundreds of nodes.

## 1.3 Clustering

One addition to multicast protocols for improving scalability is clustering, or hierarchical architecture, which is essential to achieve basic performance guarantees in large-scale MANETs. Under a cluster structure, nodes are placed into groups by assigning them different roles such as clusterhead (CH), gateway (GW) or ordinary (ON) member. Typically, an entire multi-hop MANET is divided into a number of clusters which are all independently controlled and dynamically reconfigured as nodes move. Each node within a cluster performs a different function depending on its role. A CH or group leader normally serves as a local coordinator for its cluster as it performs intra-cluster transmission arrangement, forwards data and may be responsible for route maintenance within its own group and between other clusters. A GW maintains inter-cluster links so it can access neighboring clusters and promote information sharing between them. The other cluster-members within the CH transmission range are usually called ordinary nodes. The cluster architecture improves the scalability by reducing the number of mobile nodes participating in some routing algorithm, which in turn significantly reduces the routing-related control overhead. Other advantages are less chances of interference via coordination of data transmissions and more robustness in the event of node mobility by judicially selecting stable nodes as CHs and GWs.

## 1.4 Motivation

Opinions are generally divided on the question of how to set up clusters. It can be asserted that five major objectives influenced published clustering schemes, namely forming a dominating set, having low-maintenance, performing load-balancing, and being mobility-aware or energy-efficient. Depending on their objective, proposed schemes sometimes make assumptions that may not be applicable in actual scenarios or sometimes exploit solely one side of critical tradeoffs like stability and energy-balancing.

Some energy-efficient clustering algorithms are more effective in networks with a large population and high density, but fail to consider mobility and are only suitable for low-mobility environments such as sensor networks. In some mobility-aware clustering algorithms, critical nodes perform extra work and can easily become single points of failure as they die early because of excessive energy consumption. The death of mobile nodes due to energy depletion may cause network partition and communication interruption.

It is important to balance the energy consumption among the mobile nodes to avoid node failures, especially when the nodes running out of power bear special tasks. The extra energy consumption is caused by data and control packet overhead from both the clustering algorithm and multicast protocol.

There are many proposed wireless multicast routing protocols, but there is no standard. Some of the proposed protocols are susceptible to routing loops. Routing loops cause packets to be transmitted continuously in a loop, degrading the network and causing packets to never be delivered. Some protocols break down in high mobility MANETs. The rapidly changing network topology can cause some protocols to overwhelm the network with control packets as routes are rebuilt. Some routing protocols require the use of GPS (Global Positioning System) in order to function, which is not always available.

## 1.5 Problem Statement

In a MANET nodes have to account for the lack of an infrastructure by routing data packets between themselves. Figuring out how to route data packets from one node to another, or group of other nodes requires the use of control packets. These control packets are overhead and lower the efficiency of the network. Data packets are transmitted along routes discovered by control packets to get from the source to one or multiple destinations. These routes may not always be optimal causing data packets to be transmitted unnecessarily or redundantly.

Existing multicast protocols use different techniques to try to reduce the amount of overhead required to successfully route data packets. Clustering techniques allow fewer nodes to participate in the routing function which reduces control packet overhead. Current solutions may perform well with specific applications. However, finding an optimal cluster management process and multicast protocol that works well in all situations still remains a burning issue. This led us to study the key attributes essential to provide a scalable and energy-efficient protocol for multicast routing.

## 1.6 Contributions

In our research, we focused on hierarchical multicasting with the aim of reducing control and data packet overhead. We propose (1) a new clustering technique, called Restful Stability based Insomnious Distributed Sensors (RSIDS) [10], to reduce control overhead; and (2) a new multicast protocol, called Centered and Robust Multicast Routing in Mobile Ad Hoc Networks (CPUMA) [11], to reduce data packet overhead.

RSIDS [10] is a clustering technique that takes both the remaining power of network nodes and the stability of a node within its cluster to account for mobility. RSIDS takes advantage of the benefits of prior algorithms and addresses their shortcomings. Mobility Based Metric for Clustering (MOBIC) [1] uses active clustering, which creates additional control packets and consumes greater amount of energy than the Passive Clustering (PC) algorithm [2]. In PC however, CHs and GWs work more and lose power faster than other nodes causing an "early die" problem. Geographically Repulsive Insomnious Distributed Sensors (GRIDS) [3] reduces this problem by using nodes that contain the greatest amount of energy as the critical nodes. It changes the status of nodes based on energy but does not take into account the stability of the nodes in the cluster. As the CH and GW statuses of nodes change, forwarding routes have to be recreated resulting in increased control packets. Our algorithm addresses these problems by making use of passive clustering to eliminate additional control packets, opportunistic rest periods for critical nodes to eliminate the "early die" problem, and makes use of the Stability metric to reduce the recreation of forwarding routes.

CPUMA [11] is a mesh-based multicast routing protocol that centers the core of the mesh to be close to the source nodes. It implements a distributed algorithm to periodically re-elect the core node. We center the mesh using hop count data gathered from data packets, instead of using GPS or any pre-assignment of cores to groups. Multicast data packet forwarding is directed toward the nearest mesh member to increase robustness. Nodes on the periphery of the mesh do not needlessly rebroadcast data packets received from nodes closer to the core. Our protocol reduces latency, traffic and data packet overhead compared to the PUMA [7] protocol.

This thesis is presented in article format. The first conference paper *Energy and Mobility Aware Clustering Technique*, was accepted to IEEE WCNC 2009. The second conference paper, *Centered and Robust Multicast Routing in Mobile Ad Hoc Networks*, is submitted to IEEE Globecom 2009.

## 1.7 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 contains an overview of state of the art clustering algorithms and multicast protocols. Chapter 3 presents (in article format) the first contribution of this thesis: a novel clustering technique that takes into account the energy and mobility of the network nodes. Chapter 4 presents (in article format) the second contribution of this thesis: A novel mesh-based multicast protocol that centers the core of the mesh to be close to the source nodes. Finally, Chapter 5 concludes this work and presents future research.

# Chapter 2: State of the Art

Without an existing infrastructure, nodes in wireless ad hoc networks rely on each other's participation to forward packets and communicate. Routing is enabled by the use of control packets to discover paths from any one node to another or a group of other nodes. Networks come in all sizes and densities, and as the number of nodes participating in the routing of packets increases so does the accompanying control packet overhead.

Existing clustering algorithms attempt to solve the problems of multicasting in mobile ad hoc networks utilizing various techniques to limit the number of routing nodes. In this chapter, we review clustering, its benefits and costs. Clustering techniques using dominating sets and low maintenance clustering are assessed. We review a selection of clustering algorithms in the literature that, like our own algorithm, use mobility, energy and resting periods, and influenced our research.

Second, we review the general categories of proactive and reactive multicast protocols as well as tree-based and mesh-based protocols. Achieving the maximum possible efficiency/robustness trade off in multicast routing is an area of ongoing research. We examine some of the positives and negatives of existing protocols.

## 2.1. Clustering: Definitions

Clustering is an important research topic for mobile ad hoc networks because clustering makes it possible to guarantee basic levels of system performance, such as throughput and delay, in the presence of both mobility and a large number of mobile terminals. A large variety of approaches for ad hoc clustering have been presented, whereby different approaches typically focus on different performance metrics.

Some of these approaches to clustering in MANETs are: Dominating Nodes Set - based clustering aim at finding a (weakly) connected dominating set to reduce the number of nodes participating in route search or routing table maintenance. Low-Maintenance clustering provides a cluster infrastructure for upper layer applications with minimized clustering related maintenance cost. Mobility-Aware clustering uses mobile nodes' mobility behavior for cluster construction and maintenance assigning mobile nodes with low relative speed to the same cluster to tighten the connection in such a cluster. Energy-efficient clustering avoids unnecessary energy consumption or balancing energy consumption for mobile nodes in order to prolong the lifetime of mobile terminals. Load-balancing clustering distributes the workload of a network more evenly into clusters by limiting the number of mobile nodes in each cluster in a defined range.

Motivation for implementing hierarchical routing algorithms is that they tend to be more scalable, due to their intrinsic characteristics. Many approaches have been proposed to leverage the problem of scalability (the multicast service can scale up both vertically in terms of the group size and horizontally in terms of the number of groups). The MAODV algorithm scales in scenarios with high amounts of sender nodes, but does not scale in scenarios of high mobility, high traffic load, high group size, or high number of multicast groups. The ODMRP algorithm scales in scenarios with high mobility and high group size, but does not scale in scenarios that contain high amounts of sender nodes, or high number of multicast groups [8].

In a clustering scheme the mobile nodes are divided into different virtual groups, and they are allocated geographically adjacent into the same cluster according to some rules with different behaviors for nodes included in a cluster from those excluded from the cluster. It has been shown that a flat structure exclusively based on proactive or reactive routing schemes cannot perform well in a large dynamic MANET [9]. In other words, a flat structure encounters scalability problems with increased network size, especially in the face of node mobility at the same time. This is due to their intrinsic characteristics. The communication overhead of link-based proactive routing protocols is O(n2), where n is the total number of mobile terminals in a network. This means that the routing overhead of such an algorithm increases with the square of the number of mobile nodes in a MANET. For a reactive routing scheme, the RREQ (Route Request) flooding over the whole network and the considerable route setup delay become intolerable in the presence of both a large number of nodes and mobility.

## Benefits of Clustering

A cluster structure, as an effective topology control means, provides at least three benefits. A cluster structure facilitates the spatial reuse of resources to increase the system capacity. With the non-overlapping multi-cluster structure, two clusters may deploy the same frequency if they are not neighboring clusters. Also, a cluster can better coordinate its transmission events with the help of the CH residing in it. This can save much resources used for retransmission resulting from reduced transmission collision. Another benefit is in routing, because the set of CHs and GWs normally form a virtual backbone for inter-cluster routing, and thus the generation and spreading of routing information can be restricted to this set of nodes. Last, a cluster structure makes an ad hoc appear smaller and more stable in the view of each mobile terminal. When a mobile node changes its attaching cluster, only mobile nodes residing in the corresponding clusters need to update the information. Thus, local changes need not be seen and updated by the entire network, and information processed and stored by each mobile node is greatly reduced.

## Costs of Clustering

The cost of clustering is a key issue to validate the effectiveness and scalability enhancement of a cluster structure. By analyzing the cost of a clustering scheme in different aspects quantitatively (any drop in packet reception for example), its usefulness and drawbacks can be clearly specified. Maintaining a cluster structure in a dynamically changing scenario often requires explicit message exchange between mobile nodes. When the underlying network topology changes quickly and involves many mobile nodes, the clustering related information exchange increases drastically. Frequent information exchange consumes considerable bandwidth and drain mobile nodes' energy quickly so that upper-layer applications cannot be implemented due to the inadequacy of available resources or the lack of support from related mobile nodes.

Some clustering schemes may cause the cluster structure to be completely rebuilt over the whole network when some local events take place, e.g. the movement or "death" of a mobile node results in lots of CH reelection and re-clustering. This is called the ripple effect of re-clustering. It indicates that the re-election of one CH may affect the structure of many clusters and may significantly affect the performance of upper-layer protocols in a negative way.

## 2.2. Clustering: Schemes

### Mobility Based Metric for Clustering (MOBIC)

MANETs consist of mobile nodes, and their mobility is the primary factor affecting topology changes and the need to recreate routes. In order to form stable clusters, it is important to take mobility into account. By clustering mobile nodes that are flocking together, the intra-cluster links can become more tightly connected. Re-routes and re-clustering would be decreased naturally. MOBIC selects mobile nodes that are close and are staying close to their neighbors to be CHs [1].

The aggregate of the variance of a mobile node's speed relative to its neighbors is the metric that MOBIC uses. A low value of this metric indicates a node is less mobile with regards to its neighbors and should therefore be granted CH duties.

MOBIC includes a cluster head contention timer that allows two CHs to pass each other for a short period of time before one of them takes over both clusters. This mechanism reduces incidental contact of two CHs causing re-clustering. Once a node has been deemed a CH it remains one unless its duties are taken over by another CH or the node dies. This means that the cluster is only guaranteed to have the node with the lowest mobility metric as the CH during cluster creation. Mobility is ignored after initial cluster creation. It is therefore not suitable for long lasting networks with mobile nodes of varying speed.

## Passive Clustering (PC)

Passive Clustering (PC) does not use dedicated control packets or signals for clustering. It uses data traffic forwarding to construct and maintain the cluster architecture. In PC, a mobile node can be in one of the following four states: initial, clusterhead, gateway, and ordinary node. All the mobile nodes are with 'initial' state at the beginning. Only mobile nodes with initial state have the potential to be CH. When a potential CH with initial state has something to send, such as a flood search, it declares itself as CH by piggybacking its state in the packet. Neighbors can learn the CH status by monitoring the 'cluster state' in the packet, and then record the CID and the packet receiving time. A mobile node that receives a claim from just one CH becomes an ON, and a mobile node that hears more claims becomes a GW. Since PC does not send any explicit clustering-related message to maintain the cluster structure, each node is responsible for updating its own cluster status by keeping a timer. For example, when an ON does not receive any packet from its CH for a given period, its status reverts to 'initial' [2].

PC performs well in a high mobility network where cluster topology changes frequently. PC is immune from increased control overhead due to frequent changes in network topology. It is however dependent on traffic for it to function. PC experiences problems maintaining the cluster structure in low traffic networks or networks that experience intermittent periods of high traffic followed by low traffic. In those scenarios the cluster will not be ready to route and forward data.

## Geographically Repulsive Insomnious Distributed Sensors (GRIDS)

GRIDS (Geographically Repulsive Insomnious Distributed Sensors) builds upon Passive Clustering by an efficient node status selection algorithm of critical and non-critical nodes without the periodic maintenance requirement. It adds the status and remaining energy of a node to the sent packet which are then used by receiving nodes to select node statuses. Node status is limited to clusterhead, clusterhead ready, gateway, ordinary and initial status. GRIDS inherits many advantages from Passive Clustering: it does not require any protocol dependent control packet. Well distributed insomnious nodes are guaranteed in any density of sensor networks. GRIDS is especially useful when there is a data sink which polls sensor information periodically [3].

GRIDS uses the Number of CHs and the Number of GWs to determine the next status of a node when a status change condition is met. If the Amount of CHs is greater the node will change to a GW, otherwise it becomes an ON. In GRIDS a CH and clusterhead ready node will trigger a status change upon hearing from a CH with greater energy. An ON and a GW will trigger a status change upon hearing from any neighbor node that changes its count of CH and GW neighbors. Ordinary nodes and GWs will also change status to initial status if they do not hear from a CH in time defined by a Cluster Head Timeout. Initial status and clusterhead ready status are temporary and quickly replaced. Initial status is replaced upon hearing from a neighbor and Cluster Head Ready status is replaced upon sending or hearing from a neighbor. A CH and GW node act similarly in that they forward packets and therefore consume more energy than ONs which is the only status that does not forward packets.

## Interval-Based Load Balancing Heuristics (ILBH)

ILBH (Interval-based Load Balancing Heuristic) is a heuristic to be used in conjunction with PC (Passive Clustering), which tries and reduce the overall energy consumption so that it extends the lifespan of the network. To avoid the "early die" problem of critical nodes, CHs and GWs should have a preventive mechanism which avoids critical nodes from consuming all their limited battery power. The primary function of the mechanism is to force critical nodes which served a certain period to change to 'ordinary' state and to prevent those nodes from being elected as critical node right away. In addition to the clustering status policing, ILHB switches the nodes from their active state to another "sleep" state where their energy consumption is minimal [4].

ILHB defines two thresholds $\alpha$*battery (battery represents the capacity of the node battery) and $\beta$*battery according to the local information of node batteries ($0< \alpha < \beta <1$). When a node has not reached the first threshold and after the second threshold (reformulate) (i.e., energy consumed so far is smaller than $\alpha$*battery), it operates normally as in original PC. The proposed heuristic consists of (a) changing the state of a node to 'sleep' and maintaining this state during [T1, T2] where T1 is the time the energy consumption of the node reaches $\alpha$*battery and T2 is the time the energy consumption reaches $\beta$*battery; and (2) decreasing the node's listening and reception time between the interval [T1, T2] to reduce energy consumption. These changes will allow only the nodes that have not reached $\alpha$*battery (or have reached $\beta$*battery) to become CH or GW nodes executing original PC; indeed, a node in 'sleep' state cannot change its state until its consumption reaches $\beta$*battery.

Intuitively, the proposed heuristic will allow 'balanced' distribution of energy consumption; it forces nodes with high energy consumption to 'sleep' and nodes with low energy consumption to forward data by becoming CHs or GWs. It is obvious that the proposed heuristic will generate more control traffic (e.g. when used with AODV); when a node reaches the first threshold, it changes to 'sleep' state (e.g. from CH) and extra-traffic is generated to re-cluster the network (e.g., to determine an alternative path in AODV). Simulations [4] show that even with this overhead the network lifetime is considerably increased when using ILHB.

## 2.3. Multicast Routing: Definitions

Since not all nodes in a MANET can communicate with each other directly it may be necessary for a data packet to be forwarded multiple times before it reaches its destination. Routing protocols perform the function of selecting which nodes will forward packets along to their destination. Multicast routing protocols may be proactive or reactive. A proactive protocol will try to maintain routes to all possible destinations, so when a packet needs to be sent or forwarded, the route is already known. Periodic control packets are broadcasted to announce and maintain routes. Reactive protocols search for routes only as needed. Reactive protocols have intermittent bursts of control packets instead of the steady flow of proactive protocols.

Most multicast routing protocols are Meshed-based or Tree-based. In a mesh-based protocol nodes in a multicast group form a mesh in conjunction with the nodes between them. Data packets are transmitted by all mesh members and can reach their destination via more than one path. If one path is broken, other paths are still able to deliver the multicast data packet. Control packet overhead is low, but data packet overhead is since mesh-based protocols trade efficiency for increased robustness from redundant packet delivery. Protocols that use the senders to maintain the mesh have the disadvantage of requiring multiple control packet floods per multicast group. One or more multicast group members can also be selected as the group leader to maintain the mesh avoiding the multiple control packet flood disadvantage.

Tree-based protocols forward data packets along a single route. They offer efficiency and have less data packet overhead than mesh-based protocols. However, tree-based approaches trade this efficiency for the lack of robustness in dynamic environments. Highly mobile nodes wreck havoc on tree structures and require constant maintenance.

## 2.4. Multicast Routing: Protocols

### Multicast Ad Hoc On-Demand Distance Vector (MAODV)

MAODV is a well know multicast routing protocol that creates a bi-directional shared tree for each multicast group connecting senders and receivers. Loop freedom is ensured by the use of group sequence numbers. The first node to join a group in each multicast tree is the group leader. Receivers join the tree by broadcasting a route request join packet which is responded to with route reply join packet when a tree member is found. Nodes along the path to the source node that received RREQs add routing entries when they receive RREPs creating the forwarding path.

The freshest route with the highest sequence number and least amount of hops to the tree is grafted to the tree by a multicast route activation packet [5]. MACT packets are forwarded activating the branch until they reach a node on the tree. The group leader transmits group hello packets to maintain the tree. If a link breaks, the node furthest from the group leader will try to repair the broken link. Links are repaired only when broken, and over a period of time the tree will not be optimal. The tree becomes vulnerable to more link breaks. MAODV experiences high levels of overhead when repairing broken links in a high mobility and high traffic load scenario. The need to constantly repair the tree causes control packet overhead to overwhelm the network and degrade performance significantly. Figure 2 shows the process for Route Requests and Route Replies in MAODV.

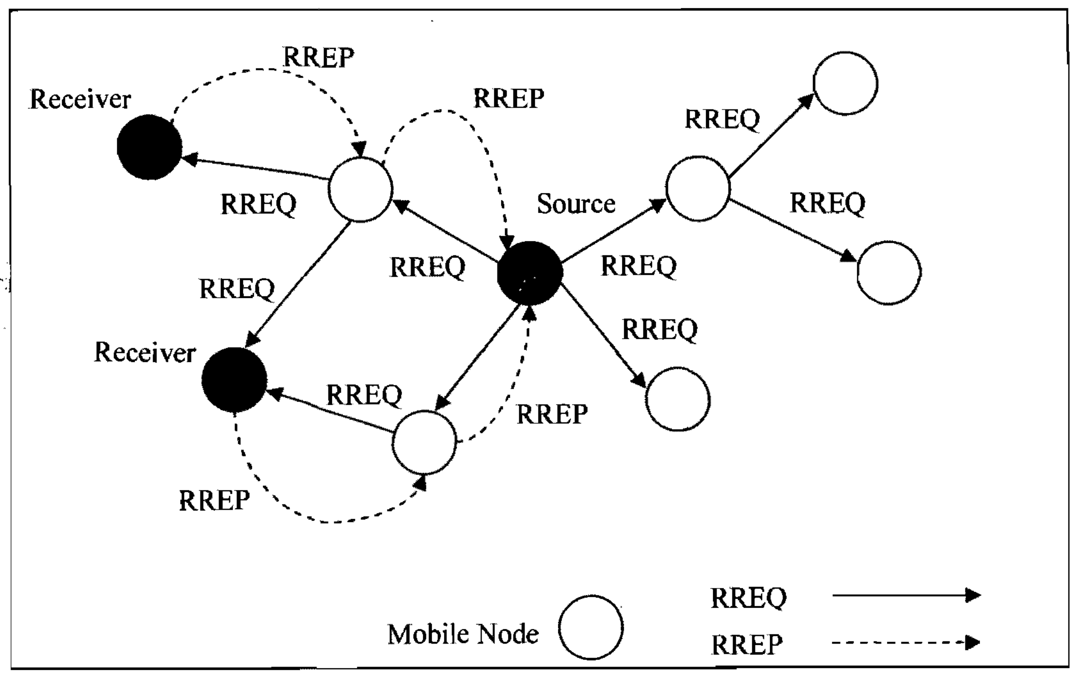


Figure 2 - MAODV Route Requests and Route Replies

## Robust Multicasting in ad hoc networks using trees (ROMANT)

ROMANT is a tree-based protocol that implements a distributed algorithm to elect one of the receivers of a group as the core of the group, and to inform each node in the network of one or more next hops to the elected core of each group. Every node has one or multiple paths to the elected core. Every receiver connects to the core along the shortest path and these paths form the tree. Senders send data packets to the group along the shortest path between the sender and the core. Once the data packet reaches a tree-member, it is flooded within the tree [6].

ROMANT uses core announcement and join announcement control packets. Core announcements contain the sequence number, address of the group, address of the core, distance to the core and the sending node address. Core announcements originate from the elected core node every three seconds and propagate to every node in the network. Join announcements contain the sender, group address and the parent of the node sending the announcement. Join announcements are transmitted every three seconds by receivers and their parent nodes along the shortest path on the tree toward the core. In this way the protocol rebuilds an optimal tree every 3 seconds. If a broken link is detected between rebuilds, nodes can use alternate next-hops. Broken links in the branches of the tree result in packets being lost. This is a liability of all tree-based protocols.

## Protocol for Unified Multicast Announcements (PUMA)

PUMA is a mesh-based protocol that evolved from ROMANT and simplified control overhead packets to just a single type: the Multicast Announcement (MA). The MA contains a sequence number, the address of the core, the distance to the core, a mesh member flag, and a parent address which is the preferred next hop from a node toward the core. The MA is used to elect cores, join and leave the mesh, update the mesh and allow nodes outside of the mesh find routes toward the core. MAs emanate from the core and are periodically transmitted. Each node that receives the MA, waits for a short period of time and transmits its own MA. These MAs transmitted throughout the network create connectivity lists that store one or more routes from each node to the core [7].

Cores are elected by a distributed algorithm that selects the highest node id claiming core status. A node without a route to a multicast group core declares itself as the core and transmits a MA to its neighbors. Once a core is chosen, it remains the core unless the network is partitioned or the core fails. The neighbors propagate the best received MA, considering a high node ID better than a low node ID. Each receiver connects to the core along all the shortest paths between it and the core forming a mesh with all the nodes along the shortest paths to the core. The parent address field in the MA allows non-members to forward multicast data packets towards the core.

### Multicasting on Directional Antennas (MODA)

MODA is a mesh-based protocol that evolved from PUMA. It makes use of directional antennas to reduce data packet overhead. It does this by using GPS to set the core at the center of the mesh and covering two hops instead of one when forwarding data packets. Each sender tries to forward data packets to a node two hops closer to the core of the mesh. Once the core node receives the data packet it makes multiple transmissions in different directions to reach nodes two hops away from it. Nodes one hop away also receive the transmission but do not need to rebroadcast the data packet lowering data packet overhead [8].

Nodes are expected to know their location and the locations of their neighbors as well as the locations of the neighbors of their neighbors. However, the algorithm requires nodes to use GPS to know these locations which is not always available or may be non-functional inside structures.

## 2.5. Conclusion

In this chapter, we reviewed existing clustering algorithms and multicast protocols for MANETs. In the literature, several clustering algorithms were proposed; they use node mobility or remaining energy as the primary factors in cluster construction and maintenance, but not both. The energy aware algorithms used different methods to distribute energy consumption but did not pay much attention to the forwarding route reconstruction required when promoting or demoting critical nodes. Several routing protocols were also proposed that balanced efficiency and robustness but left room for improvement in both areas and did not focus on reducing latency beyond pro-active routing. An attempt to center the mesh was proposed, but it required the use of GPS.

Table 1 is an outline of the properties of the clustering algorithms reviewed. The clustering algorithm (RSIDS) proposed in chapter 3 uses highest energy for clusterhead contention to maintain high energy clusterheads, stability for critical node promotion to reduce the recreation of forwarding routes, rest periods to distribute energy consumption and does not require explicit control packets for clustering.

Table 2 is an outline of the properties of the multicast routing protocols reviewed. The multicast routing protocol (CPUMA) proposed in chapter 4 uses a mesh structure for robustness, pro-active routing to lower routing delays, multiple paths to the core to reduce packet loss, forwarding paths that send data packets toward the mesh to reduce latency and improve robustness, a single control packet type to reduce overhead, source node hop counts to center the core without GPS, and has no need to repair broken links.

| Clustering Algorithm | Clusterhead Contention | Critical Node Promotion | Rest Periods | Explicit Control Packets for Clustering |
|---|---|---|---|---|
| MOBIC | Lowest Mobility | No | No | Yes |
| Passive Clustering | Highest ID | No | No | No |
| GRIDS | Highest Energy | Yes | No | No |
| ILBH | Highest ID | Demotion | Yes | No |

Table 1 – Clustering Algorithms

| Routing Protocol | Structure | Route Request | Paths to Core | Forwarding Path | Control Packet Types | GPS Required | Repair Broken Links |
|---|---|---|---|---|---|---|---|
| MAODV | Tree | On Demand | One | Tree Node | Four | No | Yes |
| ROMANT | Tree | Pro-Active | Alternate | Toward Core | Two | No | No |
| PUMA | Mesh | Pro-Active | Multiple | Toward Core | One | No | No |
| MODA | Mesh | Pro-Active | Multiple | Toward Core | One | Yes | No |

Table 2 – Multicast Routing Protocols

# Chapter 3: Energy and Mobility Aware Clustering Technique for Multicast Routing Protocols in Wireless Ad Hoc Networks

## Background

Since multicast ad hoc networks lack a physical infrastructure, they rely on nodes to participate in the routing of data packets. To enable routing, control packets are used to discover paths from the sender node to one or more receiving nodes. As networks grow in number of participants, control packet overhead increases significantly. Clustering is a method used to limit the nodes engaged in the network routing function and thus lower overhead. In this chapter, we address the issue of control overhead in MANETs by proposing a new clustering algorithm called Restful Stability based Insomnious Distributed Sensors (RSDIS).

RSIDS uses a combination of stability-based metrics, passive clustering techniques, rest periods and energy balancing techniques to create and maintain the cluster. It creates a stable cluster hierarchy that lowers control overhead and extends network lifetime.

In the simulations we compare the total and periodic packet delivery ratio, overhead, and remaining node energy of RSIDS with existing clustering algorithms. This work is presented here in article form, titled "Energy and Mobility Aware Clustering Technique for Multicast Routing Protocols in Wireless Ad Hoc Networks". It was accepted to the IEEE Wireless Communications & Networking Conference (WCNC) 2009.

# Energy and Mobility Aware Clustering Technique for Multicast Routing Protocols in Wireless Ad Hoc Networks

Astier Eric, Abdelhakim Hafid, Abderrahim Benslimane

Network Research Laboratory,
Université de Montréal
CP 6128 succ Centre-Ville
Montréal, QC, H3C 3J7, Canada
[information retirée / information withdrawn]

## Abstract

A number of key issues arise in the implementation of scalable multicast protocols for wireless mobile ad hoc networks (MANETs), namely energy consumption and data delivery over unstable/mobile nodes. To improve scalability of these protocols, clustering has been proposed. Clustering allows reducing the number of mobile nodes participating in multicast routing algorithms, which in turn significantly reduces the routing-related control overhead. In this paper, we propose a clustering algorithm, called RSIDS (Restful Stability based Insomnious Distributed Sensors), which considers both stability and residual energy of neighboring nodes when selecting critical nodes (i.e. cluster heads and gateways). RSIDS uses Passive Clustering (in opposition to active clustering) to form the clustering structure. The critical nodes selection enables the selection of most *stable* nodes with high residual energy as critical nodes; the goal is to minimize re-clustering (and thus re-branching for multicast protocols) that may generate considerable overhead and packet losses and increase the lifespan of the network. We show, via simulations, that RSIDS outperforms existing clustering schemes, in terms of packet delivery ratio and network lifetime, when used with the MAODV (Multicast Ad hoc On demand Distance Vector) routing protocol.

**Keywords-** *MANETs, Multicast Routing Protocols, Mobility, Cluster, Network Lifetime, Energy Aware, Stability. MAODV.*

# I. Introduction

Ad-hoc networks are infrastructure-less, dynamically reconfigurable wireless networks that consist of nodes that act as routers and have different power constraints. In such an environment, we are facing the problem of providing a multicast routing protocol capable of handling host mobility and the various power restrictions of the nodes.

During the last few years, several approaches have been proposed to improve multicast communication in mobile environments. Depending on how the routes connect the multicast members with each other, we can basically distinguish four categories of protocols [3]; namely Meshed-based, Tree-based, Hybrid and Stateless multicast approaches. Some protocols allow data packets to be transmitted over more than one link by creating a mesh covering all group members to increase robustness with the price of putting more redundancy in data transmission. On the other hand, tree-based approaches offer efficiency aiming at reducing the network load along with the overhead of duplicated packets and their ensuing collisions. Source or shared-tree based methods, however, lack robustness in dynamic environments. It must be acknowledged that this efficiency/robustness tradeoff raises key issues in ad hoc multicasting. Hybrid solutions aim to achieve better performance by combining the advantages of both tree and meshed-based approaches. Nevertheless, all these flat routing schemes have been shown to have limited scalability, due to their route discovery and maintenance procedures [1, 7]. Stateless multicast approaches focus on small multicast groups only.

To improve scalability of these protocols, clustering has been proposed. Clustering allows reducing the number of mobile nodes participating in routing algorithms (including multicast routing), which in turn significantly reduces the routing-related control overhead. Indeed, only cluster heads and gateways (called critical nodes) forward traffic, and therefore are part of the forwarding routes which allow nodes to reach each other. We combine clustering with tree-based multicast routing protocols like MAODV [2] by allowing all nodes to be at the originating point or receiving end of a multicast tree, but only allowing the critical nodes of the clusters to make up the routing nodes connecting them. Mobility aware clustering algorithms like MOBIC [5], form stable clusters using critical nodes with low relative speed to each other in order to minimize the probability of re-clustering. However, these critical nodes perform extra work and can easily become single points of failure as they die early because of excessive energy consumption. This may cause network partition and communication interruption. Hence, it is also important to balance the energy consumption among the mobile nodes. In energy aware clustering algorithms like GRIDS [8], energy is balanced by alternating the status of critical nodes that perform extra work and non-critical nodes that have more energy left. However, this may cause additional re-clustering which increases overhead.

The usefulness of multicasting for group-oriented applications can be compromised in MANETs if we do not envision the use of a clustering scheme (a hierarchical routing algorithm) considering both nodes' mobility and residual power. To the best of our knowledge, there is no passive clustering scheme, in the open literature, which combines both metrics to overcome MANET limitations. This paper addresses the problem of designing energy and mobility aware clustering algorithm. Our motivation comes from the fact that an energy-balancing clustering algorithm is promising, when applied to a tree-based multicast routing protocol, only if we take into account the robustness (i.e. stability) of the routes. We make use of passive clustering [4] to eliminate clustering overhead (in opposition to active clustering), opportunistic rest periods for critical nodes to eliminate the "early die" problem, and a stability metric to reduce re-clustering and thus re-branching of the multicast tree structure. We evaluate the proposed clustering scheme when used with MAODV [2] to support multicast and compare it with 4 other schemes.

The remainder of the paper is organized as follows. Section 2 presents related work. Section 3 describes details of the proposed clustering scheme. Section 4 demonstrates the effectiveness of the scheme via simulations. Section 5 concludes the paper.

## II. Related Work

Mobility is a prominent characteristic of MANETs and is the main factor affecting topology changes and routes' invalidation. Thus, it is important to take the mobility metric into account in the construction of clusters in order to form a stable cluster structure. Mobility-aware clustering indicates that the cluster structure is computed based on the mobility behavior of network nodes. The basic idea is that by grouping mobile nodes with low relative speeds into the same cluster, the intra-cluster links become more tightly connected and thus the re-clustering rate naturally decreases.

In the cluster formation phase of MOBIC [5], each mobile node sends two consecutive messages to each of its direct neighbors to help that neighbors compute their relative speeds. Then, each mobile node calculates its own aggregate local mobility and broadcasts this information to its neighbors. Also, since MOBIC has an overlapping cluster structure, a mobile node may broadcast more than one cluster-related message (cluster-related status) during the cluster formation procedure. The downside is the need for extra explicit message exchanges among mobile nodes for maintaining the cluster structure. When network topology changes frequently, it results in frequent cluster topology updates, and the control overhead for cluster maintenance increases drastically. This maintenance may consume a large portion of the network bandwidth, drain mobile nodes' energy quickly, provoke collisions and congestions, and override its improvement of the network scalability and performance. Hence, it is important to reduce the communication overhead caused by cluster maintenance.

Passive Clustering (PC) [4] is a clustering protocol that does not use dedicated clustering-protocol-specific control packets; it constructs and maintains cluster architecture based on data traffic forwarding. PC is suitable for a dense network with high mobility, where mobile nodes' continuous movement greatly affects the cluster topology. This is because the cluster maintenance of PC is traffic-dependent and immune from increased control overhead caused by frequent changes of cluster structures. PC does not make use of mobility or energy metrics which leads to critical nodes using more energy shortening the network lifetime; indeed, PC suffers from the "early die problem".

GRIDS (Geographically Repulsive Insomnious Distributed Sensors) [8] builds upon Passive Clustering and extends the lifespan of the network by using an efficient selection mechanism of critical (or not) nodes. GRIDS enables balanced energy consumption among the network nodes. Each node determines being insomnious or not based on its residual energy and the number of neighbouring insomnious nodes and their energy level.

In GRIDS, an energy abundant node can challenge cluster head and usurps the role. GRIDS uses the number of Cluster Head neighbors and the number of gateway neighbors to determine the next status of a node when a status change condition is met. However, the frequent status changes of critical nodes forces forwarding routes (and thus multicast tree structures when used with a multicast protocol) to be recreated.

In [9] the authors propose a heuristic (called ILBH and a derivative 3-ILBH) that can be used with PC in order to balance energy consumption of the network nodes; the goal is to balance energy among nodes and thus to extend the network lifetime. Two thresholds $\alpha$*battery_capacity and $\beta$*battery_capacity ($0< \alpha < \beta <1$) are defined so that when a node reaches the first threshold, it changes its state to "sleep" until it reaches the second threshold. During this time interval, the node decreases its listening and reception time to balance energy consumption and becomes an Ordinary Node. This compulsory status change will force the reconstruction of clusters and thus of multicast trees.

MOBIC uses active clustering, which creates additional control packets and consumes greater amount of energy than PC. However, in PC, Cluster Heads and Gateways work more and lose power faster and therefore die earlier than other nodes causing an "early die" problem. GRIDS eliminates this problem by using nodes with the most residual energy as the critical nodes. It changes the status of nodes based on energy but does not take into account the stability of the nodes in the cluster. 3-ILBH helps balancing energy consumption but can also force forwarding routes to be reconstructed when CHs and/or GWs are forced to sleep. In this paper, we propose an algorithm that addresses the problems of existing clustering schemes by making use of passive clustering to eliminate additional control packets, opportunistic rest periods for critical nodes to eliminate the "early die" problem, and a stability metric to reduce the recreation of forwarding routes. When used with MAODV it reduces re-branching the multicast tree structure required to deliver multicast data packets. We also make use of a derived version of 3-ILBH in order to avoid unnatural status changes and preserve established routes and thus multicast trees.

## III. RSIDS Protocol Description

In this Section, we present the details of the proposed clustering scheme called RSIDS (Restful Stability based Insomnious Distributed Sensors). A node can be in one of the following 5 states: Initial, Cluster Head (CH), Cluster Head Ready (CHR), GateWays (GW), or Ordinary Node (ON). A CH is a node that is the center of a cluster of nodes with a radius the length of the farthest node that can still receive packets from it. A GW is a node that can communicate with multiple CHs. ON is a node within the cluster that is not a CH or a GW. An Initial node is a node that has not heard from any neighboring nodes. A CHR node is a node that has not heard from any CHs and is ready to send a message. These last two states are temporary.

CHs and GWs can both forward packets, while ONs do not. This leads to CHs and GWs using more energy than ONs due to their increase use as forwarding route nodes. It is important to reduce the amount of energy spent in order to prolong the life of the network nodes and the ability of the network to communicate. To achieve a lower level of energy consumption, we use passive clustering instead of active clustering to create the network clusters. This eliminates the need for additional control packets and completely eliminates the maintenance phase of active clustering which produces additional hello packets.

### *A. Stability Metric*

To compute the stability of a node nd, we start by finding out whether a neighbor nb is coming closer or moving away from nd. The distance d to nb can be estimated as $d \cong \frac{1}{\sqrt{\left(\frac{Rx\,\mathrm{Pr}}{Tx\,\mathrm{Pr}}\right)}} = \frac{\sqrt{(Tx\,\mathrm{Pr})}}{\sqrt{(Rx\,\mathrm{Pr})}}$ where RxPr and TxPr correspond to the receiving (by nd) signal strength and transmission signal strength respectively. Friis' free space propagation model uses an inverse-square dependence of the ratio of received and transmit power on the physical distance between the transmitter and the receiver. An exact calculation of the distance may not be possible due to the difficulties measuring the transmission signal strength (involves accurate channel modeling). However, the ratio of RxPr from two successive packet transmissions can determine whether nb is moving closer or farther to nd. The relative mobility of nb relative to nd is defined as $M_{nb}^{rel}(nd) = 10\log_{10}\frac{Rx\,\mathrm{Pr}_{nb\to nd}^{new}}{Rx\,\mathrm{Pr}_{nb\to nd}^{old}}$ where $Rx\,\mathrm{Pr}_{nd\to nb}^{old}$ is the strength of the previous signal received and $Rx\,\mathrm{Pr}_{nd\to nb}^{new}$ is the strength of the latest signal received by nd from nb. If the newer RxPr is greater than the older RxPr, then the formula yields a negative value that indicates nd and nb are moving apart. If the newer RxPr is smaller than the older RxPr, then the formula yields a positive value which indicates nd and nb are moving closer to each other.

Stability S is defined as the variance to zero E of the set of relative mobility values of all the neighbors of nd: $S = \sum_{All\,nbs}\left[\left(M_{nb}^{rel}(nd)\right)^2\right]$ [5] . The lower the value of S is, the greater the stability of a node. The objective is to select critical nodes that have neighbors that remain close to them or are moving toward them. Each mobile node sends 2 delta messages to measure the relative speed with its neighbors. We use these values to calculate the stability of the node with regards to its neighbors. The result of the calculation is then broadcast to its neighbors in one or more overlapping clusters $m$. The message complexity of clustering for N nodes is therefore $O((2\Delta+1+m)N)$.

### *B. Cluster Formation*

Table 1 shows the transitions of the possible status changes of the nodes. The goal is to select high energy CHs and stable GWs. All nodes maintain a soft-state (i.e. expires) list of CHs and GWs that they can overhear. A Node starts Initial and becomes CH Ready if they hear from a Node that is not a CH; otherwise, it becomes a GW. In the initial cluster creation a CH may be surrounded by GWs, but this period does not last long as a GW becomes an ON if it hears from a node with greater stability than its own; this allows demoting less stable nodes to ONs and promoting more stable nodes to GWs. A GW may also change to Initial if it does not hear from a CH for a period of time (Cluster Head Timeout).

A CH Ready node becomes a CH upon sending successfully a packet before hearing from any CH with greater residual energy. Otherwise, a CH Ready node becomes a GW if the stability of the CH it heard from is less than its own, or an ON if it is not. When two CHs get within range of each other, CH contention occurs. The CH with the greatest amount of residual energy maintains its status and the other CH becomes an ON unless it is highly stable and becomes a GW. If a CH loses its status in a CH contention and becomes an ON it remains an ON until its residual energy is greater than all of its neighbors: it is "resting".

This helps balancing energy consumption and allows an ex CH to rest opportunistically and not be able to become a CH or a GW immediately like in the Passive Clustering or GRIDS algorithms.

| If (Node Energy <= 0) |
| --- |

```
        Then    Node State => Dead

If (Node is Resting and Node Energy >= Max Neighbor Energy)    Then    Node    Stops
Resting

Switch (Node State)

  Case Initial:
        If (Incoming Neighbor Node State != Cluster Head)
          Then          Node State => Cluster Head Ready
          Else          Node State => Gateway

  Case Cluster Head Ready:
        If (Incoming Neighbor Node State == Cluster Head &&  Incoming Neighbor Node
        Energy > Node Energy)
           Then If (Node Stability < Incoming Neighbor Node Stability)
               Then     Node State => Ordinary
             Else       Node State => Gateway

  Case Cluster Head:
        If (Incoming Neighbor Node State == Cluster Head &&  Incoming Neighbor Node
        Energy > Node Energy)
        Then  If (Node Stability < Incoming Neighbor Node Stability)
             Then       Node State => Ordinary
                             Node starts Resting
             Else       Node State => Gateway

  Case Ordinary:
        If (Node Stability >= Incoming Neighbor Node Stability and Node is Not Resting)
               Then     Node State => Gateway

  Case Gateway:
        If (Node Stability < Incoming Neighbor Node Stability) Then   Node State => Ordinary

End Switch

If (Node Sends Packet && Node State == Cluster Head Ready)
  Then          Node State => Cluster Head

If (Cluster Head Timeout)
  Then          Node State => Initial
```

Table 1 - Pseudo Code

An ON becomes a GW if it is not resting and if it hears from a node with smaller than or equal stability to its own; this allows highly stable nodes to become critical nodes. An ON may also change to Initial if it does not hear from a CH for a period of time defined by a Cluster Head Timeout.

To implement the proposed cluster formation procedure, we modified the routing protocol, under consideration, PDU to include two new fields: residual energy and stability values of the sending node.

## *C. Energy Balancing*

To further balance the energy consumption among the nodes we integrated a modified version of the 3-ILBH heuristics. We consider 3-ILBH that uses 6 thresholds (α1:0.2, β1:0.3, α2:0.4, β2:0.5, α3:0.6, β3:0.7) [9] and thus allows balancing energy consumption over 3 intervals ([α1, β1], [α2, β2], and [α3, β3]). A node that reaches a consumption of αi* battery_capacity (e.g., consumes 20% of its total capacity), is forced to a "sleep" state until its energy consumption reaches βi*battery_capacity (e.g., consumes 30% of its total capacity); then, its state can change to CH, GW or ON following the PC normal operation.

In this paper, we modify 3-ILBH as follows: if a CH or GW reaches a consumption of αi * battery_capacity, it is not forced to sleep/rest; it will be put in "sleep" state until (a) it gives up its role of GW or CH (based on the operation of RSIDS); or (b) it reaches a consumption of βi * battery_capacity. In this case, the rest period starts from the time, T, when one of these conditions is satisfied and ends when the node consumes (βi- αi)* battery_capacity starting from T.

The proposed 3-ILHB modification prevents forcing CHs and GWs to give up their roles to enter "sleep" state at fixed intervals; changing from CH/GW to "sleep" state causes re-clustering/re-branching and thus overhead and data losses. Our proposal allows CHs/GWs some flexibility in when they enter the rest period and allows the critical nodes to rest opportunistically instead of at fixed intervals. Indeed, it helps reducing transitions from CHs/GWs to "sleep" state.

### *D. Clustering Example*

Figure 1 shows a clustering structure computed by the network nodes when running GRIDS. We can see that two pairs of joining clusters (C1 and C2, C3 and C4) have the choice between two possible GWs each (N1 and N2, N3 and N4). GRIDS would choose the nodes with more energy (N2 and N4), ignoring the mobility factor. This structure is clearly not stable as nodes N2 and N4 are moving fast and in different direction than the clusters. In GRIDS, the cluster structure is only governed by energy; stability is not taken into account. Figure 2 shows how RSIDS would have clustered this scenario. The chosen GW nodes (N1 and N3) have less energy but greater Stability and will remain with the cluster longer reducing the need to recreate the forwarding routes, to reconfigure the tree and therefore reducing control packets and collisions.

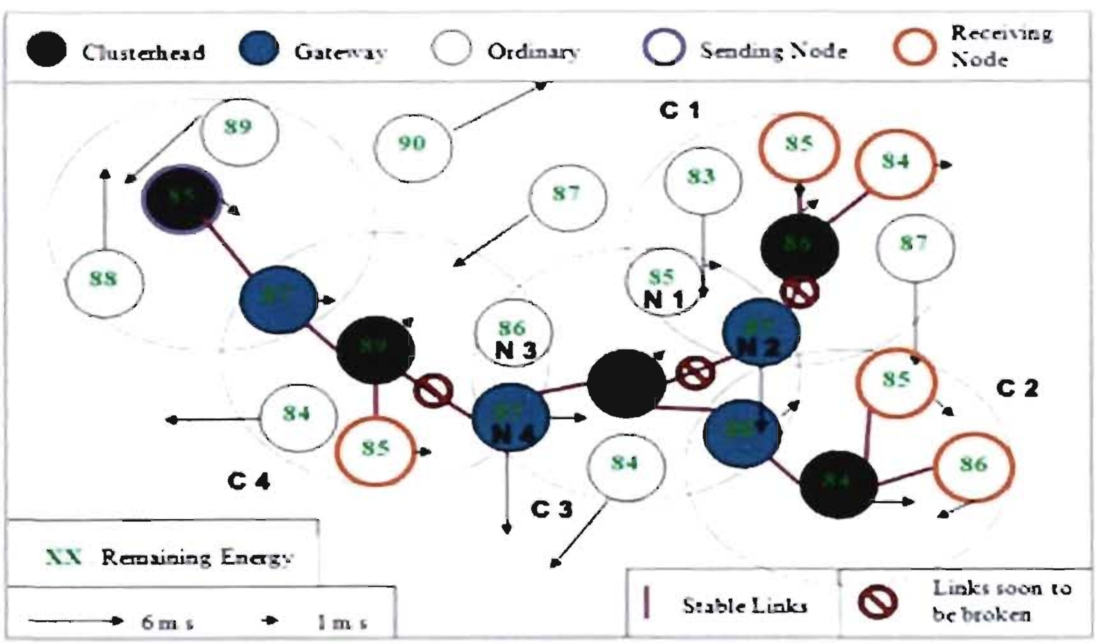


Figure 1 - GRIDS Clustering

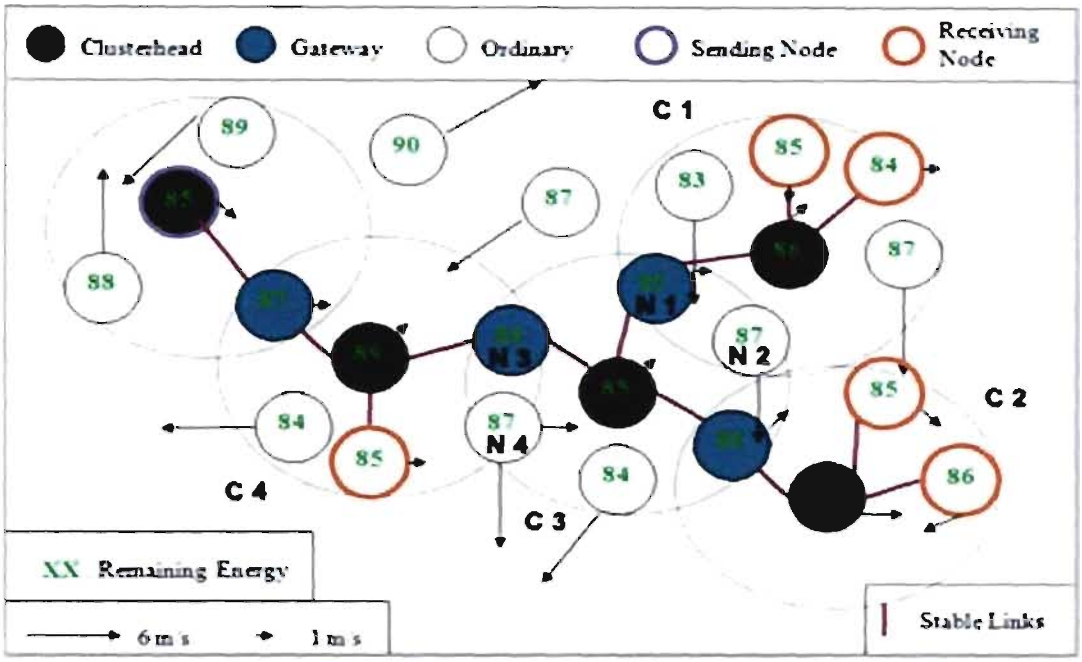


Figure 2 - RSIDS Clustering

# IV. Simulation Model and Methodology

In this Section, we present the simulation results evaluating the proposed scheme, and other clustering schemes (MOBIC, GRIDS, PC, 3-ILBH), when used by MAODV for multicast routing. We start by briefly describing MAODV and then present the evaluation analysis.

## *A. MAODV*

In MAODV, three tables are needed for routing: a Unicast Routing Table, a Multicast Routing Table and a Group Leader Table; more details on these tables can be found in [10]. When a node in MAODV creates unicast routes it sends Route Request (RREQ) packets which are responded to with Route Reply (RREP) packets when a route is found to create the forwarding route and populate the Unicast Routing Table. When a node wants to join a multicast group it broadcasts a Route Request Join packet which are responded to with Route Reply Join packets when a route to a tree member is found. After a short timeout the best route is chosen and added to the Multicast Route Table by the Multicast Route Activation (MACT) packet. The multicast tree is maintained by the group leader sending periodic Group Hello messages. When a link breakage occurs, the node downstream of the break furthest from the group leader is responsible for repairing the broken link. The clustered MAODV implementation reduces the RREQ broadcasts because any ON that receives the packet will drop the packet and not rebroadcast it. Therefore, the multicast tree can only contain ONs at the sending or receiving end of the tree, but not in the routing branches of the tree.

## *B. Simulation environment*

We used NS-2 (version 2.33) as the simulation platform. Our simulation models a MANET of 100 mobile nodes placed randomly within a 600m x 600m area. As the simulation starts each node randomly picks a new destination and travels toward it. Upon reaching the destination it randomly picks a new destination and travels toward it. Two scenarios are presented. In the first scenario, the nodes travel at a speed of 6m/s; each of 5 sender nodes sends 2 packets per second from time 30 seconds until they either run out of energy or reach the end of the simulation at 1800 seconds. The packets are Multicast CBR traffic of 512k to a multicast group of 20 receiving nodes. In the second scenario, the nodes travel at a speed of 10m/s; each of 10 Sender nodes sends to a multicast group of 10 receiving nodes. All Nodes start with 100J energy. Hellos are sent 0.75 to 1.25 seconds apart and the Cluster Head Timeout was set to 3.75 seconds, for a maximum of 3 missed hellos.

## *C. Metrics*

In our simulations, we consider the following standard metrics:

- Packet Reception Ratio: The ratio of the total number of data packets actually received versus the total number of data packets supposed to be received. This number presents the effectiveness of a protocol.

- Periodic Packet Reception Ratio: The ratio of the number of data packets actually received versus the number of data packets that were supposed to be received at the end of each 15 second interval. This number presents the effectiveness of a protocol for each interval.
- Residual Energy: The average amount of energy the nodes have at different times during the simulation. We use it to compare the energy efficiency of various protocols.
- Dead Nodes: The number of dead nodes at different times during the simulation. We use it to compare network lifetime using various schemes.

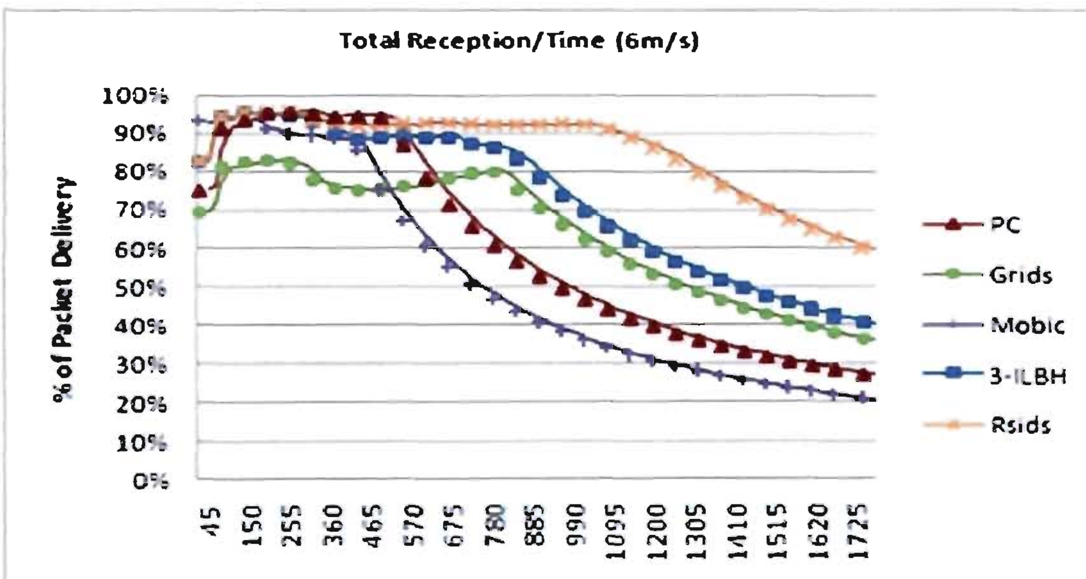


Figure 3 - Total Reception Ratio (5 to 20, 6m)

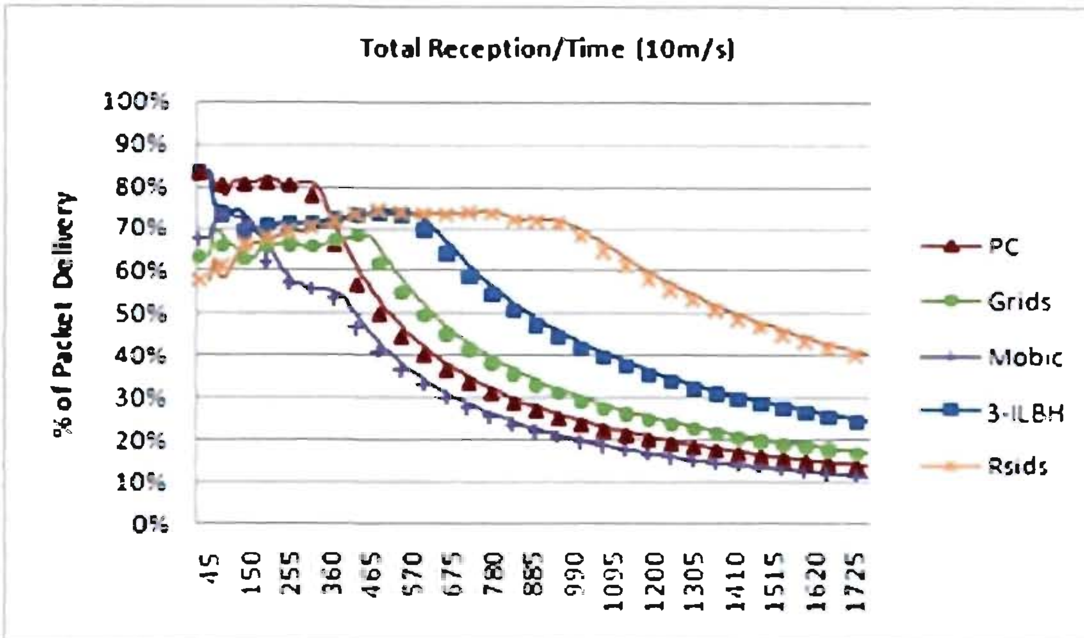


Figure 4 - Total Reception Ratio (10 to 20, 10m)

Figures 3-4 show the running total of data packet reception measured at the end of every 15 seconds over the lifetime of the simulation compared to the theoretical maximum of 200 packets/s. (2 packets/s. X 5 sending node X 20 receiving nodes.) We start at the end of 45 seconds since no data packets are transmitted for the first 30 seconds of the simulation to allow MOBIC time to cluster. We measure the number of data packets that were received per interval. It is worth noting that packets sent during interval Ii may be not counted, when computing the reception ratio, if they are received during the next interval Ii+1; this explains smaller delivery ratio values at the beginning of the simulations (Figs 3-4). At the end of the first simulation RSIDS delivers 28% more than MOBIC, 26% more than PC, 23% more than GRIDS, and 19% more than 3-ILBH.

Figures 5-6 show the periodic packet reception ratio at the end of 15 second intervals. The figures show large drops in reception for all algorithms due to the performance deterioration of the MAODV protocol when used in conjunction with multiple senders and mobile nodes [2]. In our first scenario if one sending node cannot broadcast (e.g., out of range) to the rest of the network, it can impact up to 20% of the packet reception (we have 5 senders), if one receiving node is not reachable by the rest of the network (e.g., out of range) it can affect up to 5% of the packet delivery (we have 20 receivers). When periodic reception drops off sharply toward zero, it means nodes are dying (i.e., running out of energy), which does not start for RSIDS until around 990 seconds. The last packets received for RSIDS are at 1380 seconds compared to 945 for 3-ILBH, 885 for GRIDS, 585 for PC and 495 for MOBIC. PC and MOBIC stop transmitting early because they do not take energy into account when choosing their critical nodes. Data Packet reception ends when either all the multicast source nodes/sink nodes die or when sufficient number of nodes (not belonging to the multicast group) die and no routes are available to forward data packets. RSIDS is able to maintain data reception above 80%, 11% longer than GRIDS, 13% longer than 3-ILBH, 40% longer than PC, and 52% longer than MOBIC.

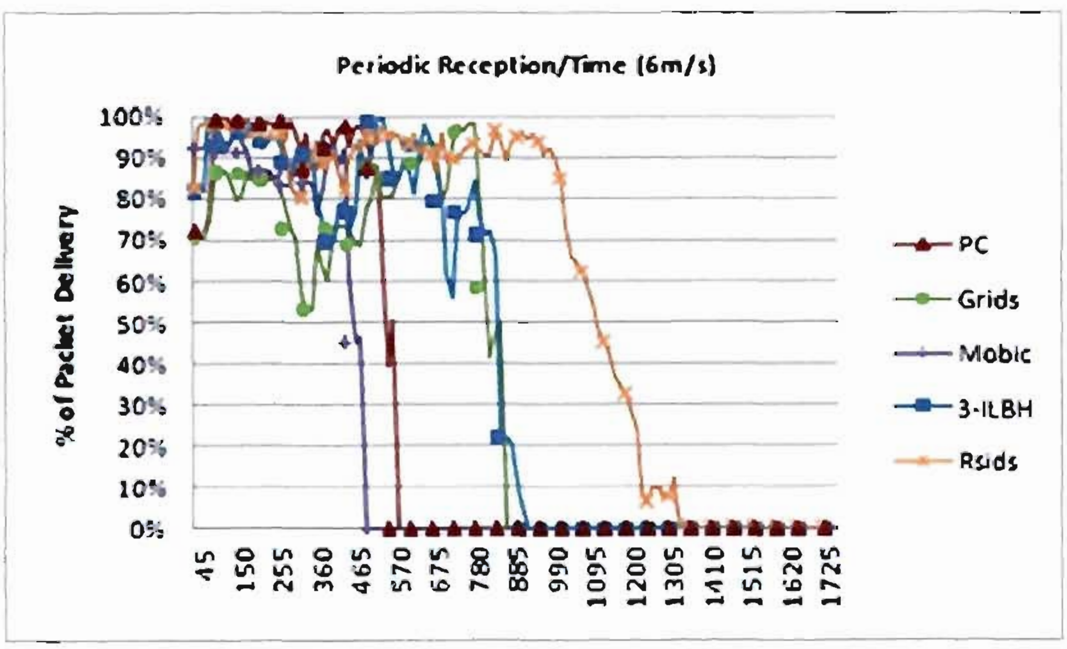


Figure 5 - Periodic Packet Reception Ratio (5 to 20, 6m)

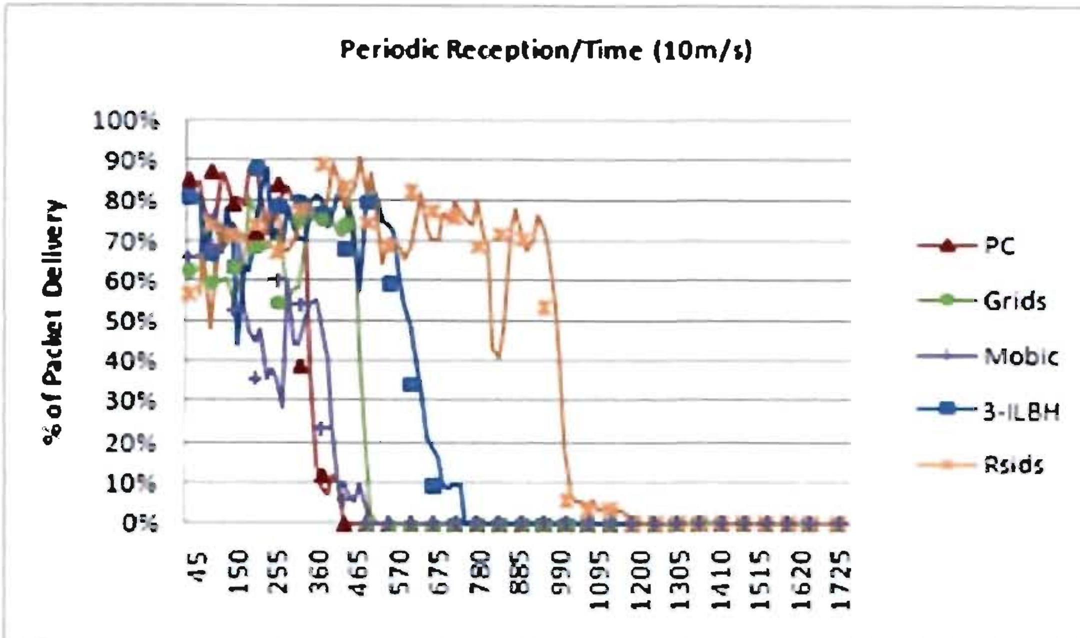


Figure 6 - Periodic Packet Reception Ratio (10 to 20, 10m)

Figures 7-8 show the average remaining energy of the nodes at 15 second intervals throughout the simulation. RSIDS was able to deliver more packets because its energy was conserved at a higher rate than the others. From the graph, the rebalancing period of 3-ILBH can be seen (if we take the derivative of the 3-ILBH curve, there are three periods where the rate of changes becomes lower: i.e. the curve is not sharply decreasing, the nodes are saving energy). We cannot observe the same tendency in RSIDS because nodes enter this period staggered by their cluster status rather than at fixed periods.

In Table 2, the first node death signifies the start of the decay in packet reception and the beginning of network death, 0% packet rate reception signifies total network death. RSIDS outlasted all others in both metrics. We use these metrics instead of absolute depletion of network node energy because once the majority of the sending and receiving nodes die, the rest of the nodes spend little energy and can survive until the end of the simulation as no data is being transmitted and their radios only send and receive occasional HELLO packets.

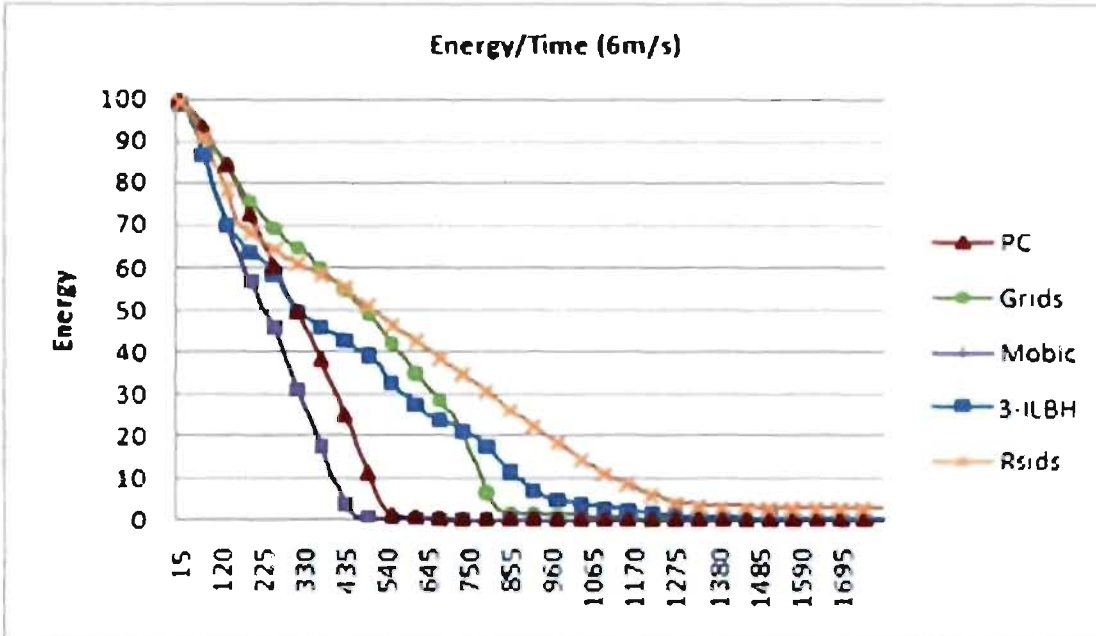


Figure 7- Remaining Energy (5 to 20, 6m)

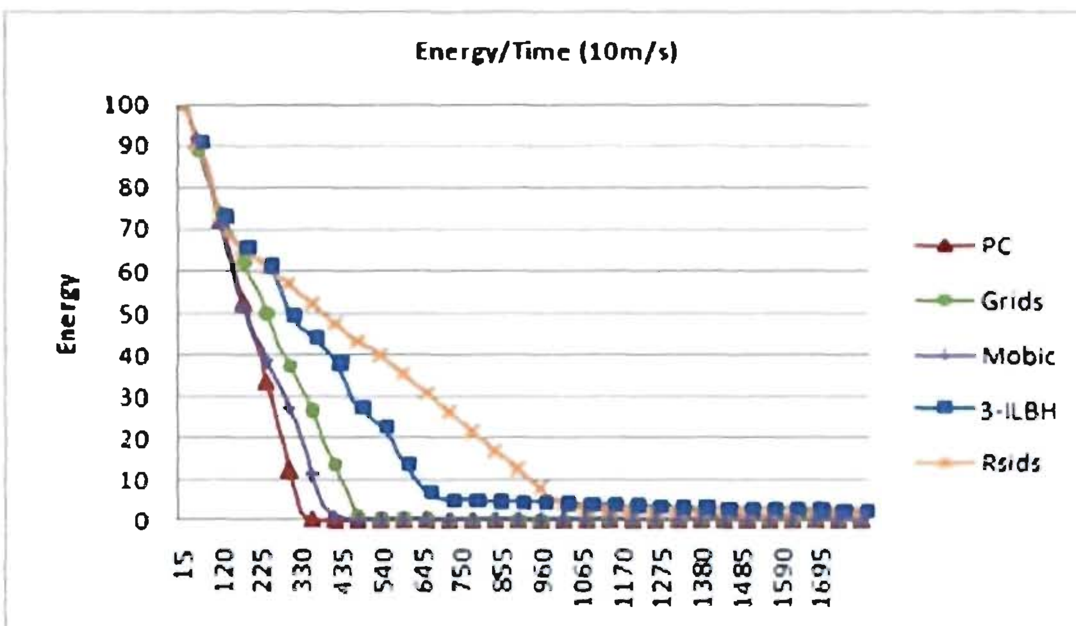


Figure 8 - Remaining Energy (10 to 20, 10m)

| 5-20, 6m/s | | | 10-20, 10m/s | |
|---|---|---|---|---|
| | 1st Node Death | 0% Reception | 1st Node Death | 0% Reception |
| PC | 525 | 585 | 330 | 435 |
| Grids | 810 | 885 | 480 | 525 |
| Mobic | 450 | 495 | 390 | 510 |
| 3-ILBH | 840 | 945 | 525 | 765 |
| Rsids | 990 | 1380 | 885 | 1245 |

Table 2 - Node Deaths and Network Death

Figure 9 shows the ratio of overhead packets (MAODV) to data packets from 30 (the start of data transmission) to the end of 435 seconds (before any of the nodes in the algorithms start dying). Using stable routes allows RSIDS to maintain a lower ratio of overhead packets to data packets. When we see a drop in the periodic packet reception ratio, it is usually correlated with an increase in the overhead ratio. This is due to the reduced amount of data packets received and the increased overhead required to find alternate routes to repair the broken multicast tree. MOBIC shows an extra use of control packets throughout due to its active clustering, and the other algorithms show higher levels than Rsids as forwarding routes break and are rebuilt. In average, RSIDS generates 18% overhead compared to 27% for PC, 29% for 3-ILBH, 30% for GRIDS, and 47% for MOBIC.

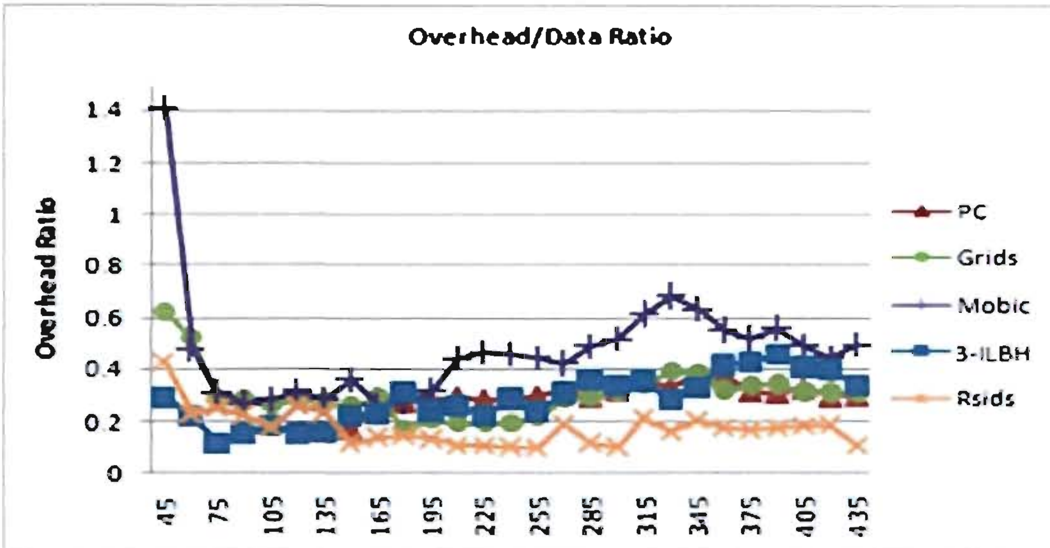


Figure 9 - Overhead/Data Ratio (5 to 20, 6m)

# V. Conclusion

We propose RSIDS, a new energy and stability aware passive clustering technique that reduces overhead by maintaining stable multicast routes. As the number of overhead packets decreases, it results in a network having less redundant/superfluous packets, having a lower probability of collisions and a less congested wireless medium. All these advantages combined with appropriate resting periods permit an increase in the network lifespan. Simulations show that RSIDS, when used to support multicast communications, outperforms existing clustering schemes. RSIDS can still be further improved; for example, since nodes know their neighbors' residual energy, they can predict when a critical node will enter a resting period/die; this will allow a suitable node to take over the critical node role at the right moment.

## References

[1] J. Wu and H. L. Li, "On Calculating Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks", *International Workshop Discrete Algorithms and Methods for Mobile Computing and Communication*, 1999, pp. 7-14.

[2] E.M. Royer and C.E. Perkins, "Multicast Ad-Hoc On-Demand Distance Vector (MAODV) Routing," Internet-Draft, draft-ietf-manet-maodv-00.txt, July 2000.

[3] C. de Morais Cordeiro, H. Gossain, DP. Agrawal, "Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions", *IEEE Network*, 2003, vol. 17, pp. 52-59.

[4] T. J. Kwon, Y. Yi and M. Gerla, "Efficient Flooding with Passive Clustering", *ACM SIGCOMM Computer Communication Review*, vol. 32, pp. 44-56, Jan. 2002..

[5] P. Basu, N. Khan and T.D.C. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks", *IEEE International Conference on Distributed Computing Systems Workshops*, 2001, pp. 413.

[6] A. B. McDonald and T.F. Znati, "Design and Performance of a Distributed Dynamic Clustering Algorithm for Ad-Hoc Networks", *Simulation Symposium*, 2001, pp. 27-35.

[7] A. D. Amis and R. Prakash, "Load-Balancing Clusters in Wireless Ad Hoc Networks", *IEEE Application-Specific Systems and Software Engineering Technology*, 2000, pp. 25-32.

[8] D. Elghnami, T.J. Kwon and A. Hafid, "GRIDS: Geographically Repulsive Insomnious Distributed Sensors – An Efficient node selection mechanism using Passive Clustering", *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, 2008, pp. 241-246.

[9] A. Hafid, F. Chender, T.J. Kwon. "Energy Aware Passive Clustering in Wireless Mobile Networks", *IEEE International Wireless Communications & Mobile Computing Conference*, 2008, pp. 535-540.

[10] Y.Zhu and T.Kunz, "MAODV Implementation for NS-2.26", Systems and Computing Engineering, Carleton University, Technical Report SCE-04-01, January 2004.

# Chapter 4: Centered and Robust Multicast Routing in Mobile Ad Hoc Networks

## Background

Mesh based multicast protocols trade efficiency for robustness by forwarding data packets via multiple links. Flooding is the extreme case where all nodes forward all data packets. The goal is to develop an algorithm that maintains high levels of delivery ratios at the lowest data packet overhead levels possible. In this chapter, we address the issue of data packet overhead in MANETs by proposing a new multicast protocol Centered Protocol for Unified Multicasting through Announcements (CPUMA).

CPUMA centers the core node of the mesh in the center of the multicast group sources. Multicast data forwarding is routed to the nearest mesh member and receivers on the periphery of the mesh only broadcast data packets when they are received from outside the mesh. CPUMA creates a robust multicast mesh that reduces latency and data packet overhead.

In the simulations we compare the total packet delivery ratio, data overhead, control overhead, latency and total network traffic of CPUMA with PUMA, one of the best existing multicast protocols. This work is presented here in article form, titled "Centered and Robust Multicast Routing in Mobile Ad Hoc Networks". It has been submitted to IEEE Globecom 2009.

# Centered and Robust Multicast Routing in Mobile Ad Hoc Networks

Eric Astier, Abdelhakim Hafid
NRL, University of Montreal, Canada
[information retirée / information withdrawn]

## Abstract

In order for a mesh-based routing protocol in a mobile ad hoc network to perform well it must achieve a high level of robustness without excessive overhead. We present the centered protocol for unified multicasting through announcements (CPUMA) for mobile ad hoc networks. A distributed core-selection and maintenance algorithm is used to find the source-centric center of a shared-mesh. We leverage data packets to center the core of each multicast group shared mesh, instead of using GPS or any pre-assignment of cores to groups, in order to reduce data packet overhead and latency. We show, via simulations, that CPUMA outperforms PUMA [5] in terms of data packet overhead, and latency while maintaining a constant or better packet delivery ratio, at the cost of a small increase in control overhead in a few scenarios.

**Key Words:** *MANETs, Multicast Routing Protocols, Mobility, Center, Core, PUMA, MAODV.*

## I. Introduction

Ad-hoc networks are infrastructure-less, dynamically reconfigurable wireless networks that consist of nodes that act as routers. In such an environment, we face the problem of providing a multicast routing protocol capable of handling high mobility, high traffic load and the ability to handle multiple sources and large multicast groups. Depending on how the routes connect the multicast members with each other, we can basically distinguish two major categories of protocols [1, 2]: Mesh-based and Tree-based protocols.

The key difference between multicast meshes and multicast trees is that in a multicast mesh data packets are transmitted over more than one path. In a mesh-based protocol, if one path is broken other redundant paths deliver the multicast message; network structure reconstruction is less frequent and produces lower control overhead. A mesh-based protocol thus benefits from an increased robustness at a cost of redundancy in data transmission and thus lowered efficiency. Existing mesh-based approaches seldom try to reduce the data packet overhead; concentrating solely on robustness. Mesh-based approaches that rely on the senders to maintain the mesh have the drawback of multiple control packet floods per multicast group. Some mesh-based approaches select one or more receivers as multicast group leaders (referred to as core nodes) to maintain the mesh and reduce network wide flooding.

In this paper, we propose CPUMA, a mesh-based protocol that provides robustness and reduces overhead, compared to existing protocols, by (1) periodically centering the core of the mesh; and (2) not allowing nodes on the periphery to rebroadcast data packets emanating from inside the mesh to reduce unnecessary data packet forwarding. Without centering the core node, receivers will form a mesh around a core node that may be at the edge of the network creating long single-use paths. The paths created to a core node centered on the sources are shorter, more robust around the area data packets must traverse and are able to reach multiple receivers.

The remainder of the paper is organized as follows. Section 2 presents related work. Section 3 describes details of the proposed multicasting protocol. Section 4 demonstrates the effectiveness of the protocol via simulations. Section 5 concludes the paper.

# II. Related Work

MAODV is a well known tree-based protocol that maintains shared-tree for each multicast group [3]. Trees composed of receivers and forwarding nodes are created by exchanging route requests, route replies and activating new branches by sending multicast activation packets toward the group leader via the shortest known paths. Periodic group hellos are transmitted by the group leader to announce and maintain the tree. Broken branches are detected by the failure to receive transmissions from a neighbor node closer to the group leader and trigger route reconstruction. In a high mobility scenario, link breaks are to be expected and cause the tree to be in a constant state of reconstruction. In a high traffic load scenario, hello packets are lost due to collisions which yield "apparent link breaks" and triggering route reconstruction [4]. This constant reconstruction results in the flooding of control packets further exacerbating the problem.

ROMANT is a tree-based protocol that solved the problem of fixing broken links in MAODV by avoiding it altogether and instead reusing the group hellos to reconstruct the group periodically [4]. Receivers periodically transmit a "join" announcement with the next hop toward the group leader gathered from the reception of group hellos. Nodes, not part of the tree, that receive these announcements and are the next hop join the tree and transmit their own announcements toward the group leader to form a tree. The protocol rebuilds an optimal tree every 3 seconds and if a broken link is detected between rebuilds via the lack of an implicit acknowledgement, nodes can use alternate next-hops. However, like other tree-based protocols, broken branches result in packets being lost.

PUMA can operate as a tree or a mesh-based protocol. It evolved from ROMANT; it uses a single type of control packet, called Multicast Announcement (MA) [5]. The MA is used to elect cores, join and leave the mesh, update the mesh and allow nodes outside of the mesh finding routes toward the core. Cores are elected by a distributed algorithm. A node without a route to a multicast group core declares itself as the core and transmits a MA to its neighbors. The neighbors propagate the best received MA, considering a high node ID better than a low node ID. Each receiver connects to the core along all the shortest paths between it and the core forming a mesh with all the nodes along the shortest paths to the core. A "parent" field in the MA contains the address of the neighbor closest to the core. A non-member forwards multicast data packets if it is the parent of the sending node. Once a core is chosen, it remains the core unless the network is partitioned or the core fails. The static core election does not take into account the distance from the core to the sources or node mobility. This can result in considerable data packet overhead because a core at the "edge" of a mesh, away from source nodes, will have excessive links and delays.

MODA is a protocol that evolved from PUMA with the aim of reducing data packet overhead. It does this by using GPS to set the core at the center of the mesh and make use of directional antennas [6]. However, GPS is not always available, and there are other ways to determine the "center" of a group of nodes.

Various distributed center location algorithms have been proposed to approximate the minimal-cost tree spanning all members of a multicast group [7]. The knowledge requirements of such algorithms include the source list, members list, and distance information. Factors to determine which node should be the center include: the maximum distance, the average distance, and the maximum diameter to the members, the sources, or all nodes in the network. However, these algorithms all use additional control overhead to elect a centered core.

The nodes in PUMA do not know all of the members of the multicast group, but the mesh members of the multicast group do know all of the sources from the broadcasted data packets. Much like MAODV failed to leverage Group Hellos, PUMA fails to leverage the knowledge gained from the data packets and selects the highest receiver id as the core of the mesh. The core in PUMA is left to wonder the network and create a non-optimized mesh structure.

CPUMA, we propose in this paper, uses the source and hop count information retrieved from data packets to calculate mesh members average hop count distance to the sources. It elects and maintains the core node in the source-based center of the multicast group mesh, and selectively rebroadcasts data packets to reduce data packet overhead and latency. It uses a single control packet type and does not significantly increase control overhead.

# III. CPUMA

## A. Overview

CPUMA is a mesh based protocol that implements a distributed algorithm to elect and maintain one mesh member (not necessarily a receiver) as the core of the multicast group. Periodic Multicast Announcements originated at the core, and broadcasted to every node in the network contain all the information needed to enable the protocol to function. Every receiver connects to the elected core along the shortest routes, and these nodes form a mesh. A sender analyses the MAs it receives and sends a data packet to the group along the shortest path to the nearest mesh node (not necessarily the core). When the data packet reaches a mesh member, it is flooded within the mesh, and nodes maintain a list of sources and the shortest hop count from the source. This data is obtained from the data packet and CPUMA header. This information is used by each mesh member to calculate the average minimum distance to the sources. The average minimum distance is simply the sum of the smallest hop counts to each source divided by the number of sources. This calculation is referred to as the weight of the member with respect to being the center of the mesh. The mesh member with the lowest weight is elected as the core. A Mesh member will periodically monitor its weight and if it is lower than the current core, it will elect itself the new core.

## B. The Multicast Announcement

The functions performed by CPUMA allow nodes to join and leave the multicast group, participate in core election, inform all nodes of their distance to the core, distance to the mesh, and next hop toward the mesh. Each node can calculate its distance to the core of the multicast group, its distance to the nearest mesh member in the multicast group. To realize these functions, CPUMA makes use of multicast announcements; these announcements are first broadcasted by the core and then altered and rebroadcasted by each recipient. A MA (Table 1) includes the following fields:

- Core ID: The address of the elected core
- Core Weight: The weight of the elected core id
- Group ID: The address of the multicast group
- Sequence number: The sequence number in the latest MA received for that group
- Parent: The address of the next hop toward the core if the current node is a mesh member otherwise the next hop toward the nearest mesh member.
- Distance to Core: One plus the distance to the core of the neighbor in the connectivity list of this group with the smallest distance to core
- Distance to Mesh: Set to Zero for all Receivers and Mesh members, otherwise one plus the distance to mesh of the neighbor in the connectivity list of this group with the smallest distance to the mesh

| Core ID | Core Weight | Group ID | Sequence Number | Parent | Distance to Core | Distance to Mesh |
|---|---|---|---|---|---|---|

Table 1 - CPUMA Multicast Announcement Format

Table 1 shows the format of each CPUMA Multicast Announcement. MAs from multiple multicast groups are aggregated together, eliminating the need for multiple MA broadcasts for each multicast group.. Table 2 shows the structure of the CPUMA Header. The CPUMA Header is included in all packets transmitted. The CPUMA header includes: (1) MA Count: Number of MAs contained in control packet (zero if data packet); (2) Hop Count: Number of times data packet has been forwarded (zero if control packet); and (3) Reserved [1]: Empty, for future use. After the CPUMA Header, the packet may contain 1 or more MAs if it is a control packet or the data being transmitted if it is a data packet. CPUMA does not combine MAs and data together in one packet.

| | | |
|---|---|---|
| MA Count | Hop Count | Reserved[1] |

Table 2 - CPUMA Header

## C. Connectivity and Source Lists

Every node in the network maintains a connectivity list using the MAs it receives from its neighbors. An element in the connectivity list contains the neighbor ID, MA reception time, and all the values of the fields in the MA are stored as received. A node will use the connectivity list to build its own MA. This list is updated with the highest sequence number announcement from each neighbor for each group and the time received. The sequence number is generated by the core node and incremented every time it sends a periodic MA. If a node receives a MA for a known group with a new core, it deletes the current connectivity list for that group and creates a new connectivity list with the first element that includes the values of the fields in that MA. The connectivity list allows a node to find the neighbor with the smallest distance to the mesh, smallest distance to the core and its multicast parent. The node chosen as the multicast parent depends on the status of the current node. It is the next hop along the shortest route to the core if the current node is a mesh member. Otherwise, it is the next hop along the shortest route to the nearest mesh member.

Every member in the multicast group also maintains a source list. The source list contains the multicast group id, the source id, and the last packet id received from that source, all extracted from the data packets of each source. The time the last data packet was received as well as the hopcount to the source are added to each entry in the list. The CPUMA header contains the hop count from the source, which is initialized to zero and is incremented by one every time it is forwarded. Data packets are flooded within the mesh; nodes maintain a packet ID cache to drop duplicate data packets. Mesh members update the hop count and time received of the source list before dropping duplicates. The source list keeps the smallest hopcount from duplicate data packets. Higher packet ids replace older entries. Entries older than the source timeout (e.g., 3 seconds) are not used when calculating node weights.

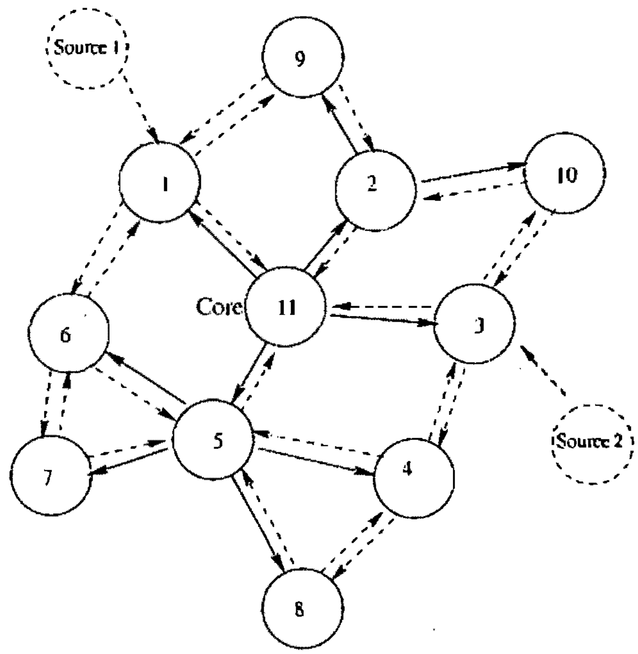


Figure 1 - Mesh broadcasting Multicast Announcements

For better understanding, let us consider Figure 1 that shows the broadcasting of MAs initiated by the core. Table 3 shows the connectivity list for node 6. The core id is 11, the group id is 224.0.0.1, the sequence number is 79 and the core weight is 1. Table 4 shows the source list maintained by node 6. The group id is 224.0.0.1 and the member weight is 3. Node 6 will transmit the following MA: MA= (Core ID: 11,Core Weight: 1 [The core, node 11 is an average of 1 hop away from both sources], Group ID: 224.0.0.1, Sequence number: 79, Distance to Core: 2, Parent: 5 [both nodes 1 and 5 could be parents, but node 1 arrived first], Distance to Mesh: 0)

| Neighbor | Distance to Core | Parent | Distance to Mesh | Time |
|---|---|---|---|---|
| 1 | 1 | 11 | 0 | 12152 |
| 5 | 1 | 11 | 0 | 12180 |
| 7 | 2 | 5 | 0 | 12260 |

Table 3 - Connectivity List at Mesh Member 6

| Source | Hop Count | Packet ID | Time |
|---|---|---|---|
| Source 1 | 2 | 309 | 12190 |
| Source 2 | 4 | 204 | 12250 |

Table 4 - Source List at Mesh Member 6

## D. Core Election and Centering

A receiver that wishes to join a multicast group from which it has not received a MA considers itself the core of that group. It starts sending periodic MAs (core_id = self, core_weight = invalid_weight, group_id = current_group, sequence_number = 1, distance_to_core = 0, distance_to_mesh = 0, parent = invalid_address) every *multicast announcement interval* (3 seconds) to its neighbors, increasing the sequence number by 1 every time.

Unless receiving a MA for a new group, or an existing group with a new core, nodes wait a short period of time before generating their own announcements. Nodes propagate MAs based on the best MAs they receive from their neighbors. A MA with a lower core weight is considered better than a higher weight; in the case of a tie, the higher core ID is considered better than a lower core ID.

The core (re-)computes its weight before sending its MA every MA interval. Every *center interval* (e.g., 15 seconds), a member of the multicast group will (re-) compute its weight and compare it to the weight of the core. If its weight is smaller than the core by the *minimum threshold* (e.g., 1 hop or 10%), it elects itself as a core, updates its MA (core_id = self, core_weight = node_weight, group_id = current_group, sequence_number = 1, distance_to_core = 0, distance_to_mesh = 0, parent = invalid_address) and broadcasts it to its neighbors. It is worth noting that our simulations did show that setting the center interval equal to the MA interval resulted in an increase in control packet overhead without significant improvements; using 15 seconds as the center interval yielded the best overall performance.

A node that receives a MA with a core id and weight that are better than it currently has will update its group connectivity list and broadcast an MA immediately. Eventually every node will receive a MA with the best core id and weight for that multicast group. If a receiver does not hear a MA for 3 times MA intervals it elects itself as the core of that group and begins transmitting MAs.

## E. Mesh Establishment and Maintenance

Receivers set their mesh distance to zero in their MAs to indicate they are mesh members. A non-receiver becomes a member if its connectivity list contains a fresh entry with at least one mesh member with a bigger hop count to the core than itself. An entry is considered fresh if it was received within 2 times MA intervals. This allows all shortest paths from the receivers to the core to be included in the multicast mesh. Nodes transmit an immediate new MA whenever their mesh distance changes to or from zero. A node outside of the mesh sets its parent to the neighbor in the connectivity list with the shortest distance to the mesh and sets its distance to the mesh as 1 plus the value of its parent's distance to the mesh. In the case more than 1 neighbor has the same distance to the mesh, the earliest received connectivity list entry is chosen.

## F. Forwarding Multicast Data Packets

The neighbors in the connectivity list with a smaller distance to the mesh are the potential next hops to the multicast group. A non-mesh node forwards a multicast data packet if it is the parent of the node that sent the data packet. Multicast data packets are forwarded hop by hop until they reach the nearest mesh member at which point they are flooded within the mesh. The packet ID cache allows nodes to drops duplicates.

When a node that is not a mesh member transmits a packet, it expects its parent to forward it. When the parent forwards the packet, the node that originally sent the packet will also hear the forwarded packet. This mechanism serves as an implicit acknowledgement that the packet was received. The connectivity list is updated and neighbors removed if a node does not receive an implicit acknowledgement of the data packet transmission within the *acknowledgement period.*

In PUMA all mesh members forward packets, and all receivers are mesh members. In CPUMA receivers forward packets only if they have mesh children or receive a packet from outside the mesh. A receiver that is a parent to a mesh member is within the mesh; it is a hop on the path from another receiver to the core. A receiver that is not a parent to a mesh member is on the periphery. There is no need for receivers on the periphery of the mesh to rebroadcast data packets received from within the mesh, since no node outside of the mesh is interested in receiving the packet. Table 5 presents the pseudo code for some of the functions of CPUMA.

| | |
|---|---|
| Node Weight | `SUM(smallest_hop_count_to_each_source) / number_of_sources` |
| Join Group | `send_multicast_announcement()` |
| Leave Group | `/* do nothing - node will timeout */` |
| Elect Core | `if (coreId == Unknown) then coreId = self`<br>`send_multicast_announcement()`<br>`/* If core is unknown, become core and send MA */`<br>`if (coreId == Unknown OR  ma.coreWeight < getGroupWeight OR (ma.coreWeight == getGroupWeight AND ma.coreId > coreId))`<br>`then coreId = ma.coreId`<br>`send_multicast_announcement()`<br>`/* if a node receives a MA and the core is unknown or the MA has a weight lower than the current multicast group weight, or the weights are equal but the Id of the core node in the MA is greater than the current core Id, then accept the new core and send a MA */` |
| Distance to Core | `if coreId == self then distance_to_core = 0`<br>`else find_smallest_distance_to_core_in_connectivy_list()`<br>`/* traverse the connectivity list and return the smallest distance to the core from all neighbors if the node is not the core */` |
| Distance to Mesh | `if (coreId == self OR i_am_a_receiver OR i_am_a_meshMember)`<br>`then distance_to_mesh = 0`<br>`else find_smallest_distance_to_mesh_in_connectivy_list()`<br>`/* traverse the connectivity list and return the smallest distance to the mesh from all neighbors if the node is not in mesh */` |
| Forward Data Packet | `if (datapacket.sentfrom.parent == self OR (i_am_a_mesh_member AND number_of_mesh_children() > 0))`<br>`then broadcast_data_packet()`<br>`/* broadcast the data packet if the node is the parent of the node sending the data packet or if the node is a member of the mesh, and has neighboring mesh nodes with greater distances to the core.  Otherwise, this node is a receiver on the periphery of the mesh and has no need to broadcast the data packet */` |

Table 5 - CPUMA Pseudo-code

# IV. Simulation Model and Methodology

In this Section, we present the simulation results comparing CPUMA with PUMA. PUMA concentrates mesh redundancy in the region of the receiver chosen as core. CPUMA concentrates mesh redundancy in the region of the mesh between the source nodes, and therefore the area where data packets must travel through. We compare both of these algorithms using NS-2[8]. We thank Sidney Doria for the PUMA code for NS-2.

To illustrate the data packet overhead savings of CPUMA we consider a simple example with 1 source, 3 receivers and static nodes (Figures 2-3); solid nodes indicate Mesh Members while dashed nodes indicate non-members. Sources and Receivers are labeled in both figures. Figure 2 shows the mesh structure after core election in PUMA. The highest receiver ID is elected Core and the other receivers connect via the shortest paths to it; the number next to each node indicates the number of times a data packet is broadcasted before it reaches that node. In PUMA the Source forwards packets toward the Core; once the core receives the packets it forwards them to both receivers. It takes a total of 9 broadcasts to reach all receivers: 4 broadcasts to get from the source to the Core (via nodes a-b-d) and 5 broadcasts to reach the receivers (1 broadcast by the core and 1 broadcast by f, g, h and i each).

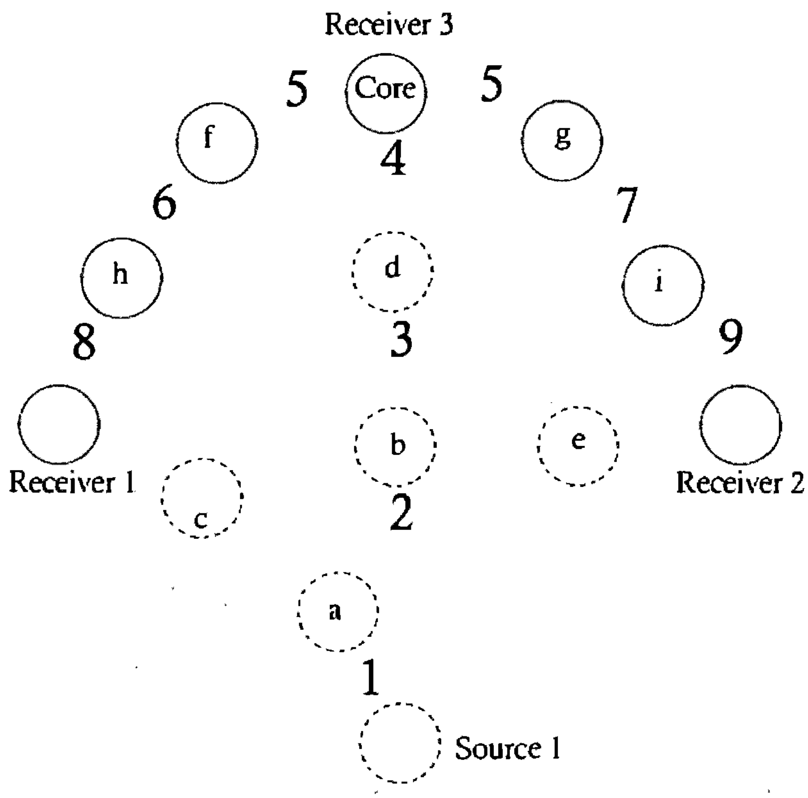


Figure 2 - PUMA showing Data Packet Overhead

Figure 3 shows the mesh structure after the second re-centering of the core in CPUMA. The Source forwards data packets toward the closest mesh member instead of the core, in this case Receiver 1 via nodes a and c. At the first re-centering of the core Receiver 1 becomes the core and the mesh is recreated. Receiver 2 finds the shortest path to Receiver 1via nodes e-b-c. Receiver 3 finds the shortest path to Receiver 1 via f-h. Node c is selected as the core in the second re-centering of the core. Receiver 3 finds a new shortest path to node c via nodes d and b and the mesh is optimized. It now only takes 6 broadcasts to reach all receivers; 2 from the source to the core, one more from the core to Receiver 1, b broadcasts and reaches both d and e which each broadcast once more to get to Receiver 2 and Receiver 3.

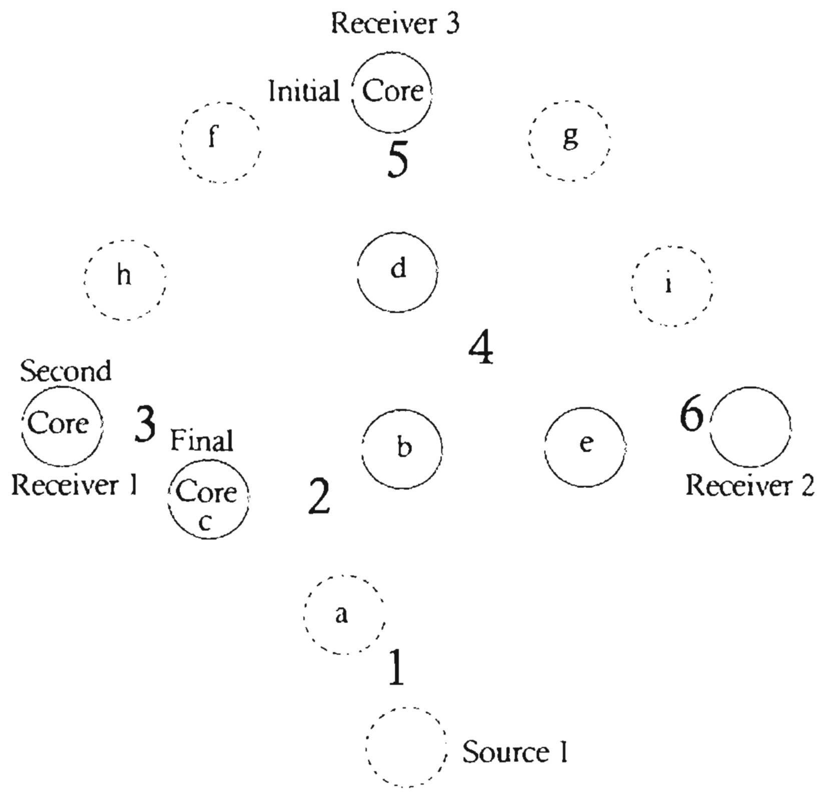


Figure 3 - CPUMA after Centering the Core

| | PUMA | CPUMA |
|---|---|---|
| **Data Packets Sent** | 5970 | 5994 |
| **Data Packets Received** | 17587 | 17936 |
| **Data Packets Forwarded** | 58686 | 30058 |
| **Delivery Ratio** | 98.20% | 99.74% |
| **Control Packets Sent** | 13357 | 13053 |
| **Latency** | 0.057 | 0.035 |

Table 6 - PUMA vs. CPUMA statistics

We simulated one source sending packets twice a second to the three receivers with zero mobility configured as shown above for 3000 seconds. As seen in Table 6, compared with PUMA, CPUMA lowers the amount of data packets forwarded by almost 50% (58686 vs. 30058) for the simple static example. The number of control packets is practically the same since only two rounds of core re-centering are performed. Delivery ratio is slightly improved and latency is improved since the number of hops from source to receivers is reduced.

## A. Metrics

The metrics used in our evaluation are packet delivery ratio, control overhead, data packet overhead, latency and traffic. Packet delivery ratio is the number of data packets delivered divided by the data packets that should have been delivered (data packets sent x number of receivers). Control overhead is the number of control packets that are generated divided by the number data packets delivered. Data packet overhead is the number of data packets transmitted divided by the number of data packets delivered. Latency is the sum of the delay between sending a packet (from the source) and receiving it (by the receiver) for all data packets divided by the number of data packets received. The data packets overhead is more important than the control overhead since the data packets are 17 times larger than the control packets (544 compared to 32 bytes). Traffic is the sum of the total Kbytes transmitted. The PUMA and CPUMA headers are equal in size, so no extra overhead is incurred.

## B. Scenarios

The values of the simulation parameters used in all experiments are shown in Table 7. Four experiments were carried out to compare PUMA with CPUMA.

| Simulation Parameters | |
|---|---|
| **Simulator** | NS-2 version 2.33 |
| **Simulation Time** | 700 seconds |
| **Simulation Area** | 1000m x 1000m |
| **Node Placement** | Random |
| **Pause Time** | 0 |
| **Mobility Model** | Random Waypoint |
| **MAC Protocol** | IEEE 802.11 – 1997 |
| **Data Packet Size** | 512 |
| **All other Parameters** | NS-2 Defaults |

Table 7 - Simulation Parameters

We used scenarios similar to those found in [5]:

1. Experiment 1: Mobility assumes 1, 5, 10, 15, and 20m/s; Senders = 5; Members = 20; Traffic Load = 10 packets/s
2. Experiment 2: Senders assumes 5, 10, 15, and 20; Mobility = 5 m/s; Members = 20; Traffic Load = 10 packets/s
3. Experiment 3: Members assumes 5, 10, 20, 30, and 40;. Mobility = 5 m/s; Senders = 5; Traffic Load = 10 packets/s
4. Experiment 4: Traffic Load assumes 10, 20, 30, 40, and 50 packets/s; Mobility = 5; Senders = 5; Members= 20

Senders and Receivers are chosen randomly from among the 50 existing nodes. Traffic load is equally distributed among all senders. A traffic load of 10 packets/s and 5 senders is divided so each sender sends 2 packets/s. R stands for Receivers, S for Senders, M for mobility and T for traffic in the graphs below.

## C. Results

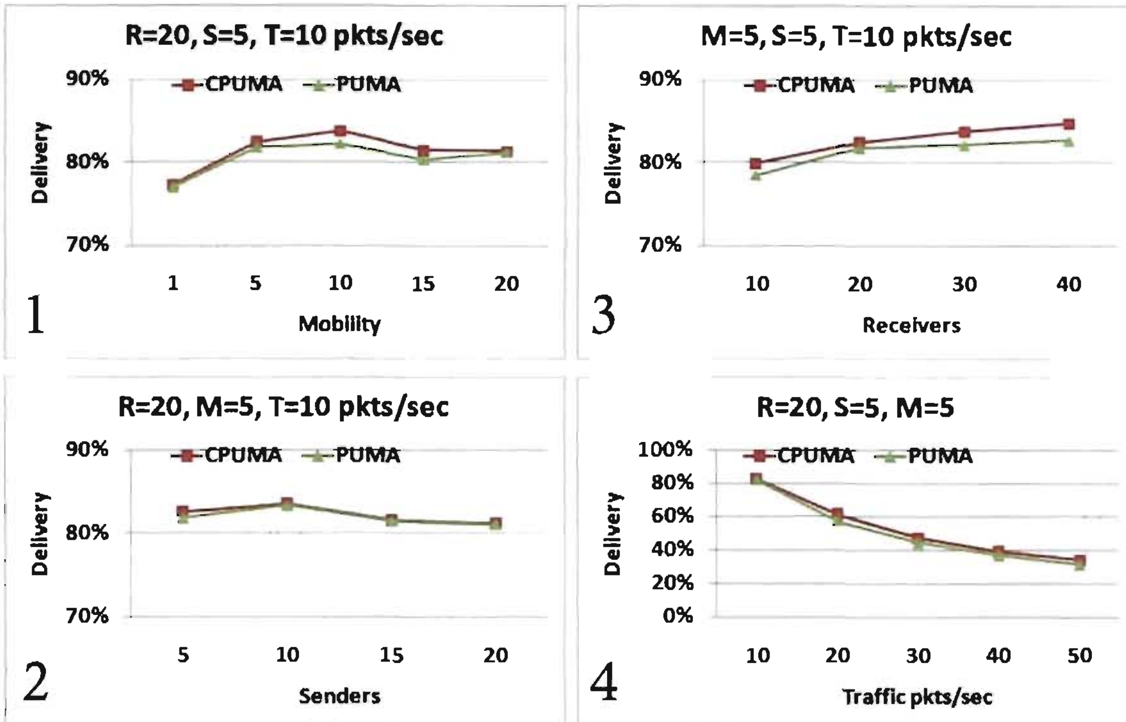


Figure 4 - Packet Delivery Ratio

Figure 4 shows a small improvement in packet delivery ratio across the board for CPUMA. Indeed, CPUMA delivers 0.3 to 2% more data packets than PUMA in most scenarios except 4-4. In 4-4, the network is very congested and the reduced packet forwarding allows CPUMA to outperform PUMA by 2-3.5%. Since data packets travel toward the nearest mesh member instead of the core, data packets benefit from the redundancy of the mesh sooner and are less likely to be lost.

Figure 5 show the control overhead of CPUMA to PUMA to be practically equal except in 5-2. In 5-2, the number of senders increases resulting in more values used to calculate node weights (in the case of CPUMA). The weight calculations change faster resulting in the frequent centering of the mesh, and therefore more control packets. The increase is small (2-2.5%) since mesh members only check their weights in 15 second intervals. If the *center interval* is lowered to 1 second, control packet overhead doubles in our 20 sender scenario without a great effect on the results. This is because the core changes around a few nodes near the current center without affecting the mesh structure. It is prudent to choose a reasonable interval so the core is not constantly changing.

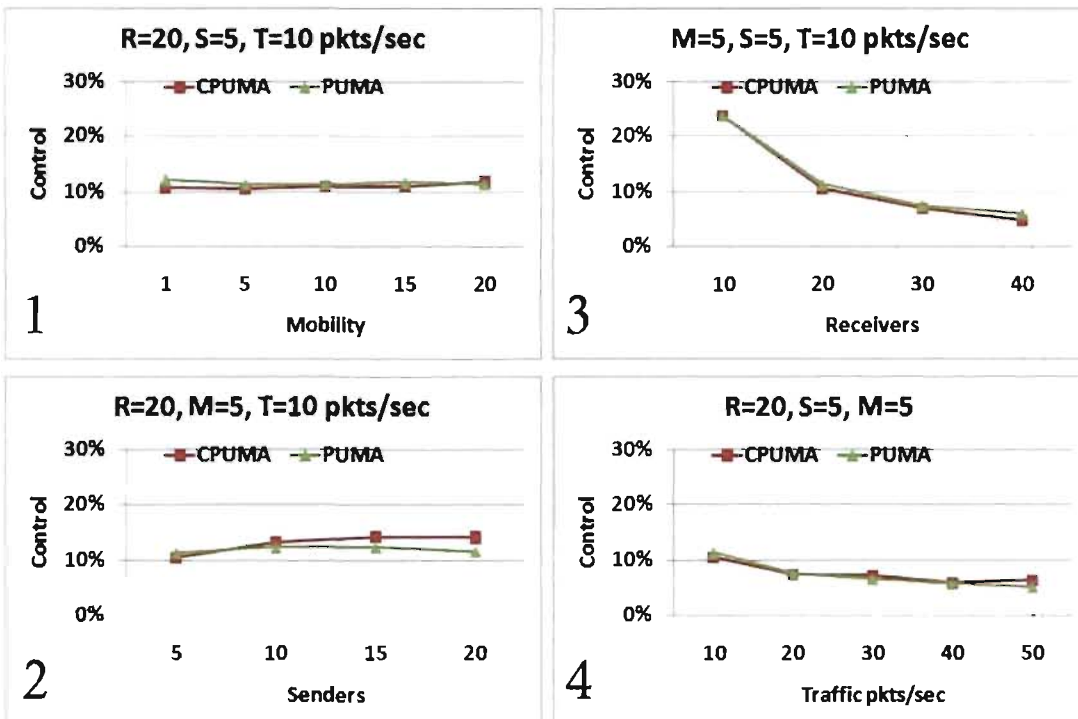


Figure 5 - Control Overhead

Figure 6 shows that CPUMA handily outperforms PUMA. Indeed, CPUMA achieves an average overhead reduction of 30%; the reduction exceeds 50% in 6-2 with 15 senders. The reduction in data packet overhead is maintained when faced with changes in mobility, the number of senders, the number of receivers and the amount of traffic. 6-3 shows a smaller improvement than the others (14% - 20%) because as the number of receivers approaches 100%, more nodes have to be included in the mesh and PUMA and CPUMA construct similar meshes.

Figure 7 shows a large difference in the latency of CPUMA compared to PUMA. The latency for CPUMA in 7-1 averaged 0.11s compared to 0.26s (more than twice as much) for PUMA. In 7-2 latency averaged 0.09s for CPUMA compared to 0.29s for PUMA (more than 3 times as much). The latency difference is more pronounced in 7-3 (4 times as much) as the number of receivers increases and in 7-4 (6 times as much) as traffic increases. This is due to nodes forwarding data packets toward the mesh, and having a mesh near the center of all of the senders in the network. In PUMA, a sender node would instead send its data packet toward the non-centered core which may be at the other side of the network. This increases the length of the path the data packet must travel before reaching the mesh and results in a longer delay reaching the receivers.

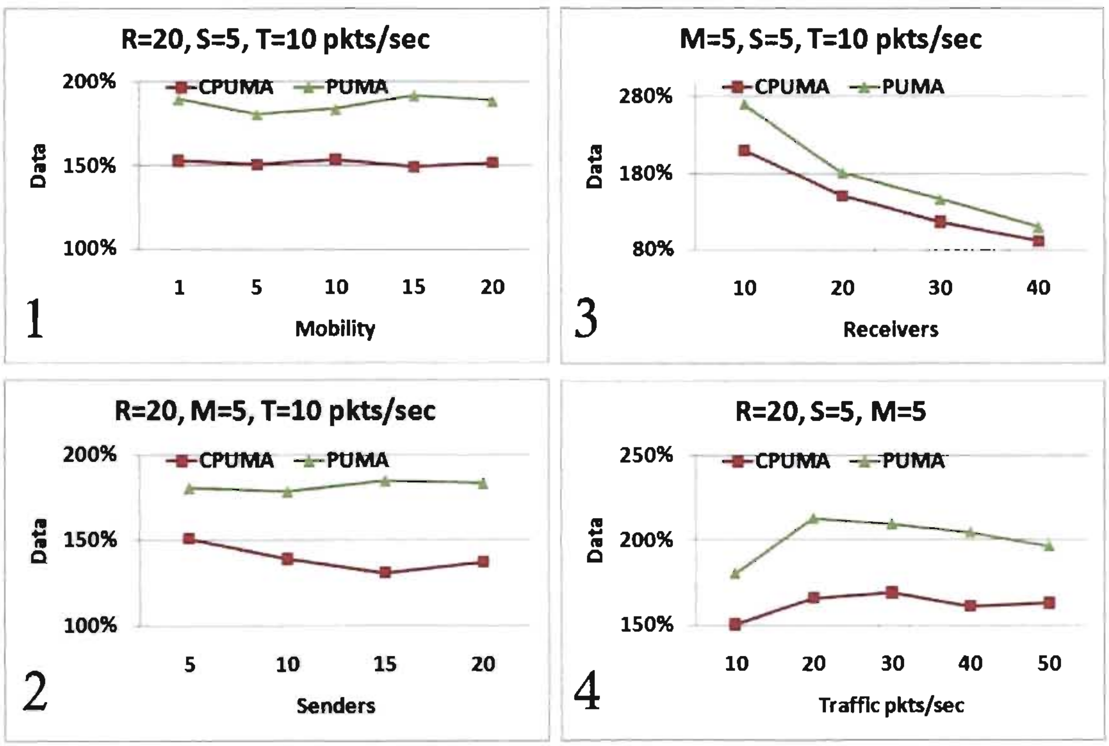


Figure 6 - Data Packet Overhead

Figure 8 shows the difference in the traffic of CPUMA compared to PUMA. All simulations show an improvement as CPUMA produced an average 18.4% less traffic than PUMA. 8-2 had the best results at an average of 21.8% less traffic which correlates to the data packet overhead reduction shown in 8-2. In 8-4 CPUMA has an average of 14.1% less traffic than PUMA but averages 2.3% higher packet delivery ratios.

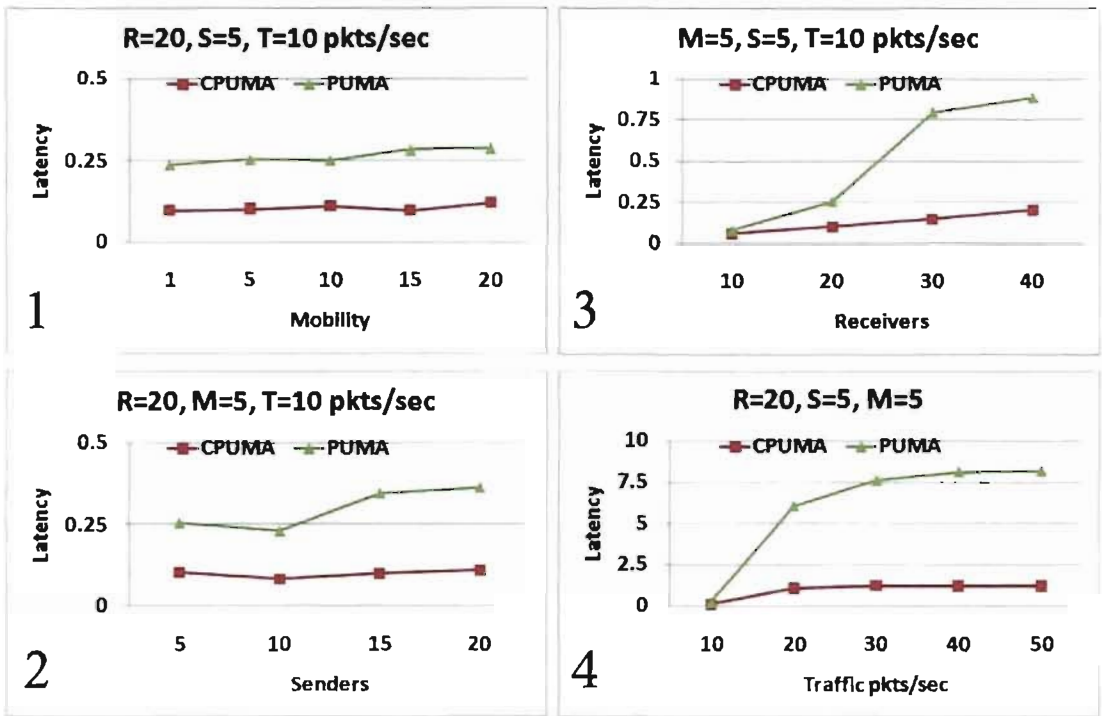


Figure 7 - Latency

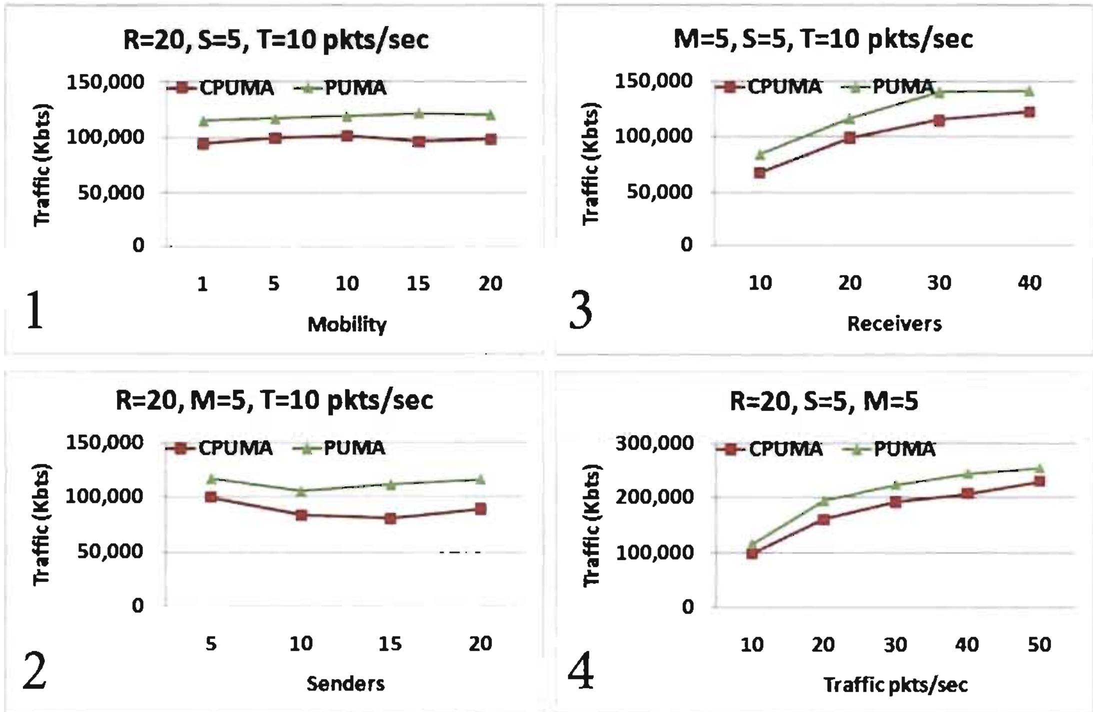


Figure 8 - Traffic (Kbytes)

# V. Conclusion

The Centered Protocol for Unified Multicasting through Announcements (CPUMA) is based on the novel idea of leveraging information gained from data packet transmissions to find the source based center of the mesh. The mesh created in CPUMA benefits from a focused redundancy in the area around the source nodes where the data packets originate and must therefore travel through in order to reach all of the receiving nodes.

Additionally a centered mesh enables forwarding routes outside of the mesh to be directed toward the nearest mesh member instead of the core reducing the latency of transmissions and taking advantage of the benefits of the mesh redundancy sooner.

Furthermore, in CPUMA mesh members selectively forward data packets only when they are certain to have a receiving node waiting for the transmission in order to reduce data packet overhead. The receiver nodes on the edge of the mesh do not have an audience and therefore have no need to broadcast data packets. The control overhead in CPUMA is not constant, but it is contained. The mesh is re-centered at timed intervals and since not every node can qualify as a new possible core node, a total mesh reconstruction is not performed. Since the new core is selected from mesh members around the previous core the re-centering function serves mostly as a simple update with very minor mesh changes.

CPUMA maintains a considerably lower data packet overhead and latency than PUMA while maintaining or improving packet delivery ratio and not significantly increasing control overhead regardless of mobility, traffic, senders or receivers in the network.

## References

[1] S.J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols", *Joint Conference of the IEEE Computer and Communications Societies*, 2000, vol. 2, pp. 565-574.

[2] C. de Morais Cordeiro, H. Gossain, DP. Agrawal, "Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions", *IEEE Network*, 2003, vol. 17, pp. 52-59.

[3] E.M. Royer and C.E. Perkins, "Multicast Ad-Hoc On-Demand Distance Vector (MAODV) Routing," Internet-Draft, draft-ietf-manet-maodv-00.txt, July 2000.

[4] R. Vaishampayan and J.J. Garcia-Luna-Aceves, "Robust Tree-Based Multicasting in Ad Hoc Networks", *IEEE International Conference on Performance, Computing and Communications*, 2004, pp. 647-652.

[5] R. Vaishampayan and J.J. Garcia-Luna-Aceves, "Protocol for unified multicasting through announcements (PUMA)", *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2004.

[6] R. Vaishampayan, J.J. Garcia-Luna-Aceves and K. Obraczka, "Multicasting On Directional Antennas (MODA)", *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2005, pp. 659-664.

[7] A. Karaman and H.S. Hassanein, "Core-Based Approach in Multicast Routing Protocols," *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, SPECTS, 2003, pp. 525-532.

[8] Breslau, Estrin, Fall, Floyd, Heidemann, Helmy, Huang, McCanne, Varadhan, Xu, and Yu. "Advances in Network Simulation", *IEEE Computer*, vol. 33, 2000, pp. 59-67.

# Chapter 5: Conclusion and Future Work

In our research we were interested in the issues of multicasting in a mobile ad hoc network. We took two different approaches. First, we created a clustering technique to limit the number of nodes involved in the routing functions of an existing multicast protocol, taking into account mobility and node energy. Second, we created a new multicast protocol with lower overhead than existing protocols by optimizing the mesh as nodes move and eliminating unnecessary data packet forwarding.

RSIDS is a clustering technique that uses energy and stability data to create a hierarchy of mobile nodes. RSIDS uses passive clustering to reduce overhead and maintain stable multicast routes. It reduces packet overhead which results in a lower probability of collisions and lowers network congestion. It also makes use of dynamic resting periods to produce an increase in the network lifespan. Simulations demonstrated that when used to support multicast communications, RSIDS outperformed existing clustering schemes. Remaining energy and mobility are two important node attributes in an ad-hoc network; taking both in to consideration when creating clusters yields better results than taking only one or the other into account.

CPUMA is a multicast protocol that lowers network latency and reduces data packet overhead. It centers the mesh based on the source nodes using information retrieved from data packets. In addition, nodes on the periphery of the mesh selectively determine if they need to rebroadcast data packets. The source nodes forward data packets toward the nearest mesh member to reduce latency and improve robustness. These features allow CPUMA to achieve lower data packet overhead and latency than existing mesh-based protocols while retaining high packet delivery rates. The reuse of the data packet information to find the source based center of the multicast group is an idea that deserves further investigation and may lead to other novel opportunities for multi-purpose packet transmissions.

RSIDS and CPUMA are both improvements over existing alternatives in ad-hoc networking and they each have their uses. As a mesh based protocol CPUMA is suited for a network of any size and medium to low density. The application layer relying on the CPUMA protocol must place an importance on high packet delivery rates from redundancy over low overhead. In a high density network the data packet delivery redundancy of the mesh will yield unnecessary overhead since the mesh will be constructed with all the shortest paths to the core, and packets will be forwarded by all mesh members.

As a clustering technique RSIDS is better suited for large dense networks in order to take advantage of overhearing and limit redundancy in the case of flooding or using a mesh based protocol. It is not well suited for small sparse networks since in that kind of scenario most nodes would end up being clusterheads or gateways eliminating the advantages of clustering altogether.

RSIDS and CPUMA are not mutually exclusive. The RSIDS clustering technique can be overlaid the CPUMA multicast protocol much like the MAODV multicast protocol. The CPUMA Header which is included in every transmission can be altered to include the transmitting node's current cluster status (Cluster Head, Gateway, or Ordinary Node). When building the CPUMA mesh, nodes that have a cluster status of Ordinary Nodes would not be allowed to be parent nodes. This change would eliminate all Ordinary Nodes from being included in the multicast forwarding routes outside of the mesh. Any Ordinary Nodes that are non-forwarding mesh members would still be counted when determining how many mesh member children nodes belong to a parent node. This change would make sure Ordinary Nodes that are Receivers would still be able to receive forwarded packet transmissions from their parent node. Ordinary nodes would be excluded from being the Core node of a multicast group and would not participate in the re-centering of the mesh.

## 5.1. Future Work

RSIDS could be further improved by adding predictive features. Nodes know their neighbor's last broadcast energy level and could use this information to predict when a current critical node will enter a resting period or die. A suitable node could take over the critical role at the right moment eliminating packet loss. It would also help reduce control packet overhead repairing the links.

CPUMA could be further improved by dynamically allocating the time between re-centering the core node. Long times between re-centering in a network of slow moving nodes would lower control packet overhead, short times between re-centering in a fast moving network would help maintain low latency and low data packet overhead which would compensate for higher levels of control packet overhead. When the re-centering interval is equal to or greater than the multicast announcement interval, control overhead should remain stable since MAs transmitted during re-centering reset the MA announcement timer. If the re-centering interval is less than the multicast announcement interval the mesh will be in a constant state of reconstruction and that would yield additional control overhead and cause a significant drop in the packet delivery rate. Since multicast data is forwarded toward the nearest mesh member instead of the core, routes outside of the mesh are not affected by core changes in CPUMA. To further study the effects of re-centering, the algorithm can modified so that nodes that neighbor the current core would not be able to become cores. Simulations could be run limiting the ability to claim core status to nodes 1, 2 and 3 hops away from the current core.

The RSIDS clustering algorithm could be paired with the CPUMA multicast protocol without a lot of changes in order to handle multicasting in large scale high density wireless ad-hoc networks.

# Bibliography

[1] P. Basu, N. Khan and T.D.C. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks", *IEEE International Conference on Distributed Computing Systems Workshops*, 2001, pp. 413.

[2] T. J. Kwon, Y. Yi and M. Gerla, "Efficient Flooding with Passive Clustering", *ACM SIGCOMM Computer Communication Review*, vol. 32, pp. 44-56, Jan. 2002.

[3] D. Elghnami, T.J. Kwon and A. Hafid, "GRIDS: Geographically Repulsive Insomnious Distributed Sensors – An Efficient node selection mechanism using Passive Clustering", *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, 2008, pp. 241-246.

[4] A. Hafid, F. Chender and T.J. Kwon. "Energy Aware Passive Clustering in Wireless Mobile Networks", *IEEE International Wireless Communications & Mobile Computing Conference*, 2008, pp. 535-540.

[5] E.M. Royer and C.E. Perkins, "Multicast Ad-Hoc On-Demand Distance Vector (MAODV) Routing," Internet-Draft, draft-ietf-manet-maodv-00.txt, July 2000.

[6] R. Vaishampayan and J.J. Garcia-Luna-Aceves, "Robust Tree-Based Multicasting in Ad Hoc Networks", *IEEE International Conference on Performance, Computing and Communications*, 2004, pp. 647-652.

[7] R. Vaishampayan and J.J. Garcia-Luna-Aceves, "Protocol for unified multicasting through announcements (PUMA)", *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2004

[8] R. Vaishampayan, J.J. Garcia-Luna-Aceves and K. Obraczka, "Multicasting On Directional Antennas (MODA)", *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2005, pp. 659-664.

[9] .Y. Yu and P.H.J. Chong, "A Survey of Clustering Schemes for Mobile Ad Hoc Networks", *IEEE Communications Surveys and Tutorials*, vol. 7, 2005, pp. 32-48.

[10] E. Astier, A. Hafid and A. Benslimane, "Energy and Mobility Aware Clustering Technique", *IEEE Wireless Communications and Networking Conference (WCNC)*, 2009.

[11] E. Astier and A. Hafid, "Centered and Robust Multicast Routing in Mobile Ad Hoc Networks", submitted to IEEE Global Communications Conference, 2009.